Video Coding Standards

AVS China, H.264/MPEG-4 PART 10, HEVC, VP6, DIRAC and VC-1

Springer



视频编码全角度详解

AVS China、H.264/MPEG-4 PART10、HEVC、VP6、DIRAC、VC-1

[美] 卡米塞提·拉姆莫汉·饶（K.R.Rao）
金道年（Do Nyeon Kim） 著
[韩] 黄在静（Jae Jeong Hwang）

刘雪冬 刘雪莲 撒继铭 等译

机械工业出版社
CHINA MACHINE PRESS

VISIT…

LANZAROTE Caliente.com

视频编码全角度详解：

AVS China、H. 264/MPEG-4 PART10、HEVC、VP6、DIRAC、VC-1

［美］卡米塞提·拉姆莫汉·饶（K. R. Rao）
金道年（Do Nyeon Kim） 著
［韩］黄在静（Jae Jeong Hwang）
刘雪冬　刘雪莲　撒继铭　等译

机 械 工 业 出 版 社

本书是一本关于多种视频编码方式标准的专著，介绍了 H. 120、H. 261、MPEG-1、MPEG-2/H. 262、H. 263 系列、MPEG-4、VP6、Dirac、VC-1、H. 264/MPEG-4 第 10 部分、H. 265/HEVC 和我国的 AVS China 等标准，以及标准之间的转码问题。本书重点阐述了标准内在的基本功能、工具、技术和操作，同时提供了大量的文献资源和研究专题。

本书适合作为学术界、研究机构和工业界相关人士的参考书。

Translation from English language edition:
Video coding standards
by K. R. Rao, Do Nyeon Kim and Jae Jeong Hwang
Copyright © 2014 Springer Netherlands
Springer Netherlands is a part of Springer Science + Business Media
All Rights Reserved

本书由 Springer 授权机械工业出版社在中华人民共和国境内（不包括香港、澳门特别行政区及台湾地区）出版与发行。未经许可之出口，视为违反著作权法，将受法律之制裁。

北京市版权局著作权合同登记　图字：01-2015-5790 号。

图书在版编目（CIP）数据

视频编码全角度详解：AVS China、H. 264/MPEG-4 PART10、HEVC、VP6、DIRAC、VC-1/（美）卡米塞提·拉姆莫汉·饶（K. R. RAO）等著；刘雪冬等译. —北京：机械工业出版社，2017. 7

书名原文：Video Coding Standards：AVS China，H. 264/MPEG-4 PART 10，HEVC，VP6，DIRAC and VC-1

ISBN 978-7-111-56826-1

Ⅰ. ①视…　Ⅱ. ①卡…　②刘…　Ⅲ. ①视频编码 – 研究　Ⅳ. ①TN762

中国版本图书馆 CIP 数据核字（2017）第 105191 号

机械工业出版社（北京市百万庄大街 22 号　邮政编码 100037）
策划编辑：吕　潇　责任编辑：吕　潇
责任校对：肖　琳　封面设计：马精明
责任印制：常天培
涿州市京南印刷厂印刷
2017 年 7 月第 1 版第 1 次印刷
169mm×239mm · 28 印张 · 2 插页 · 552 千字
0001—2500 册
标准书号：ISBN 978-7-111-56826-1
定价：149.00 元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换

电话服务	网络服务
服务咨询热线：010-88361066	机 工 官 网：www. cmpbook. com
读者购书热线：010-68326294	机 工 官 博：weibo. com/cmp1952
010-88379203	金　书　网：www. golden-book. com
封面无防伪标均为盗版	教育服务网：www. cmpedu. com

译　者　序

随着多媒体时代的到来，人们对于视频的质量要求越来越高，与此同时，也期望视频传输具有更快的速度。而传输系统和存储系统对于视频格式乃至码流语法的要求则是尽可能统一。这两方面的客观要求就是，要对视频数据进行压缩以及压缩要实现标准化。故而视频编码标准应运而生。K. R. Rao、D. N. Kim 和 J. J. Hwang 合著的本书囊括了从人类进入信息时代至今的主要视频编码标准，本书涉及的内容之广，是不同寻常的。本书最具特色，也是最有价值的部分是提供了大量的研究专题，尤其是那些具有前瞻性的尝试和质疑。这些专题可以作为硕士乃至博士论文的选题，也可作为一篇期刊论文的研究焦点。国内视频编码领域的研究生毕业论文会因为选题重复或陈旧而遭受质疑，研究者们发表论文也会因为题目平凡而被拒稿。相信广大读者在充分吸取本书的滋养、挖掘所列专题这一“富矿”之后，对于研究内容的确定是大有裨益的。此外，本书还提供了篇幅可观的附录，其中包含了与专题研究相关的实验数据结果，这些结果具有非常高的参考价值和指导意义，可以作为广大读者自行研究的参考。

在翻译过程中，考虑到翻译的目的是介绍编码技术的不同标准，而技术标准本身是客观存在的，是独立于表述语言和方式的，因此，我们对译文的要求首先是“客观”，尽可能地忠实于技术本身而不是刻板于原著的表述。其次的要求是“地道”，即符合汉语的表达习惯，阅读起来感觉流畅。对于原著中没有给出的缩略语，译者查阅了相关视频标准的文档进行了解释和补充。对于那些译者自己也不甚满意的译法，注明了原著的表述，供广大读者斟酌参考。对于图表等实验结果，则是不折不扣地“忠实”于原著。

本书由刘雪冬负责第 1 ~ 5 章的翻译、全书的统稿和校对。中国地质大学（武汉）外国语学院的刘雪莲老师负责第 6 ~ 8 章和附录 A ~ E 的翻译。武汉理工大学信息学院的撒继铭老师负责附录 F ~ H 的翻译。在本书的翻译工作中，武汉理工大学信息学院的郭志强、黄朝兵、江雪梅、张小梅、周颖、许建霞、黄铮和朱健春老师以及贾静静、霍波、宋深和张一丹同学参加了部分翻译工作。在此，感谢这些老师和同学们的大力支持！感谢机械工业出版社的相关工作人员，特别是吕潇编辑。吕编辑谦虚体谅，宽厚礼让，在译者因故致使翻译工作受阻时给予充分的理解并给予相应的时间调整，使译者最终以良好的状态如期完成翻译工作。

作为主要译者的刘雪冬是武汉理工大学信息学院的副教授。有幸来到美国得克萨斯大学阿灵顿分校在本书原著作者 K. R. Rao 教授的指导下开展访问学者的研究工作，这也是成为本书译者的机缘之一。译者初到阿灵顿时气温偏凉，Rao 教授关

心地问候译者是否需要他的衣服和围巾，甚至是否需要午饭。在以后的见面中总会提醒是否来杯咖啡，在分别时常会送一些小零食。Rao 教授常打球游泳开车，为学生授课并指导研究，身体很棒。殊不知他已 85 岁高龄！看来，“仁者寿”是不分种族国籍的。为了尽可能地翻译准确，译者曾就原著中的一些问题当面请教了他，这些问题涉及缩略语、算法和硬件实现的技术细节乃至词语的拼写，老先生对于每个问题都能立刻做出回答。目前，作为 IEEE 会士（Fellow），Rao 教授仍然关注着视频编码研究的最前沿，在他指导译者阅读的文献中，视频标准发展的图谱已经绘制到了 2017 年。心有仁善，慈爱为怀；老骥伏枥，壮心不已，焉能不寿？祝愿 Rao 教授身体健康，事业精进！

另一件让译者倍感欣慰的是本书对于我国具有自主知识产权的音视频编码标准 AVS-China 进行了较为详细的介绍，这表明中国人制定的编码标准其综合性能引起了世界的关注！祝愿祖国的科技事业繁荣昌盛！最后，要特别感谢武汉理工大学对于译者出国研修的资助，没有学校提供的经费，译者就无缘遇到 Rao 教授，更不会有此番学术和翻译经历。

译者在翻译的过程中，对原著存在的一些明显错误进行了修改。如果书中仍然存在疏忽与错误之处，恳请读者批评指正。意见或建议请发至 zxndp@126.com。

译　者

美国得克萨斯州阿灵顿市

2017 年 3 月

原　书　序

就我们知识所及，本书综合了所有最近的主要视频编码标准：AVS China、H.264/MPEG-4 Part 10（今后称之为 H.264/AVC）、VP6（目前 VP10）、Dirac、VC-1/2 以及 HEVC/NGVC（HEVC：高效视频编码/下一代视频编码），在这个意义上说，本书是不同寻常的。除了最后的标准外，所有的这些标准已被工业界（在不同层次上）采用，为大众市场开发消费电子产品（机顶盒、搜索引擎、笔记本电脑、PC、平板电脑、网络书籍、数码相机、DVD、蓝光光碟、智能手机、各种手持移动设备、视频点播、视频流等）。基于 HEVC/NGVC 的产品也处于不同的开发阶段，这一点可以从各种论文中看到（会议论文集、期刊论文、专题报告、主题演讲——见第 5 章）。本书的焦点在于，导致各种比特率下的压缩/编码、质量级别和应用场合的这些标准内在的基本的功能性、工具、技术和操作，详细的逐步实现过程是有意省略的。然而，除了标准的性能比较和局限，本书还呈现了对于这些标准的透彻理解。借助于在线资源，诸如 web/ftp 地址，标准文档、开源软件、述评论文（期刊出版物以及会议陈述）、主题演讲、专题报告、技术反思和相关资源，读者可以很便利地在不同档次（profile）和级别（level）上实现编码器和解码器以评价它们的性能。

独具慧眼又独出心裁的读者可以改善性能，降低复杂度和探索另外的应用。除了第 1 章和第 2 章，剩余的各章均以专题结束（可以调整修改做硕士论文，某些甚至可以做博士论文），这些项目在此抛砖引玉，启发思想。此外，本书展望了从一个标准到另一个标准的转码（不是一个标准的编/解码器连接另一个标准的编/解码器）。通过将一个标准的部分编码比特流有效地/高效地用于其他标准内，以此来显著降低转码器的总体复杂度。这本身就不是件容易的事情。然而，许多挑战已被克服，那就是 MPEG-2 到 H.264/AVC 转码器的实现，反之亦然，以及 H.264/AVC 到 VC-1 的低复杂度转码。

HEVC/NGVC（见第 5 章）是最新的标准，旨在与 H.264/AVC 相比，在相同视觉质量下降低大约 50% 的比特率。2013 年 1 月，它的最终国际标准草案（FDIS：Final Draft International Standard）已经面世（主要是 10bit 位深和所有的帧内档次），工作草案定期更新。这已被 ITU 批准为 ITU-T H.265/ISO/IEC 23008-2 HEVC。预计基于 HEVC 的解码器在 2016 年底将达到 20 亿之多。紧随其后的是计

划在 2014 年实现标准化的扩展部分如可伸缩视频编码（SVC：Scalabe Video Coding）和多视图编码（3D 视频，自由视点视频，立体视觉视频等）。除了已有标准如 H.264/AVC 到 HEVC 的双向转码器之外，这些进展还为其他的研发工作提供了肥沃的土壤（见最后的项目）。本书的主要焦点是视频，对音频只做少量描述。然而，某些论题/专题详细叙述了在保证视频和音频同步时编码器内的编码过程和视频/音频比特流的复用过程，以及这些比特流后来的解复用和解码过程。读者可以参考 Rao 博士的网址 http://www-ee.uta.edu/dip（单击“courses”，然后单击“EE5359 multimedia processing”，向下滚动访问 theses/projects/ppslides/papers/proposals 等），即用 AAC 音频比特流复用/解复用 AVS China 视频以实现音视频同步，以及用 HEVC-AAC 音频比特流复用/解复用 H.264/AVC 视频实现音视频同步。

第 6 ~ 8 章阐述了 VP6（最初由 On2 技术开发——后来由 Google 获得）所包含的功能，还带有下列标准相关的网址：VP9/VP10、BBC 开发的使用小波而不是传统 DCT/整数 DCT 的 Dirac，以及基于微软视窗媒体视频 9（Windows Media Video 9）的 VC-1。除第 1 章和第 2 章外，所有其他章都增补有专题，许多专题具有硕士和博士论文水平。附录 A ~ H（源于得克萨斯大学阿灵顿分校以前的研究生，附录 B 和 H 除外）提供了额外的资源，这些资源也补充了许多具有硕博士论文水平的专题。书目提要提供了数量众多的出现在第 3 章到第 8 章的有关视频编码标准的参考文献，包括 web/ftp 地址、评论文章、标准文档、主题报告、专辑、开源软件等。

希望本书和所有概述过的资源能为读者理解和实现这些标准提供可能。有远见的读者可以对这些标准提出改变或增补、或扩展、或修正以提高性能、降低复杂度和增加应用领域。本书无意成为任何课程的教科书。相反，希望本书成为学术界、研究机构和工业界的具有科研水平的参考书。

致　谢

本书是三位作者 K. R. Rao、D. N. Kim 和 J. J. Hwang 长期合作的结晶。特别感谢来自他们家庭的支持、坚持和理解。Kin 博士和 Hwang 博士都是得克萨斯大学阿灵顿分校（UTA：Universtity of Texas at Arlington）多媒体处理实验室（MPL：Multimedia Processing Laborotory）的访问教授，他们持续做出的显著贡献使本书成为可能。本书第一作者要感谢工程学院的院长 Jean-Pierre Bardet 博士、电气工程系系主任 J. W. Bredow 以及 UTA 的所有同事，感谢他们以各种形式给予的支持。还要感谢微软公司的 G. J Sullivan 博士、圣塔克拉拉大学的 Nam Ling 博士、Ankur Saxena 博士和 Zhan Ma 博士（都来自三星研究实验室）、北京大学的 Wen Gao 博士、德州仪器的 M. Budagavi 博士、华盛顿大学的 M. T. Sun 博士、Fraunhofer HHI 的 H. Lakshman 博士、BBC 的 T. Borer 博士、夏普实验室的 Deshpande 博士、Bankoski 博士和 D. Mukherjee 博士（均来自谷歌）、Interdigital 公司的 Y. Reznik 博士、佛罗里达大西洋大学的 H. Kalva 博士、和伦敦大学玛丽女王学院的 E. Izquierdo 博士，感谢他们的坚定支持。UTA 校友 Priyadarshini Anjanappa 女士为本书贡献了不计其数的时间来将所有松散的部分综合起来（著作权发布、校对、以及无数的其他细节）。该校友和 UTA 多媒体处理实验室（MPL）的研究生们以各种途径给出了建设性的意见，特别是对于第 5 章。我们按照这些意见进行了数不胜数的修订和更新。所有的附录（附录 B 和 H 除外）基本上是专题报告或者是 UTA 的 MLP 里以前的学生论文。这些学生是 B. Shrestha、C. Chukka、P. Anjanappa、S. Muniyappa、R. Veerla 和 M. Krishnan。本书作者衷心感谢他们的贡献。

目 录

译者序
原书序
致谢
缩略语

第 1 章 引言 …… 1

1.1 流行的视频和音频标准 …… 1
1.2 视频的数字表示 …… 12
1.3 视频编解码的基本结构 …… 14
1.4 视频编译码性能比较的衡量标准 …… 16
1.5 音频的数字表示 …… 23
1.6 感知音频编码的基本结构 …… 24
1.7 音频编译码的性能比较测度 …… 26
1.8 总结 …… 30

第 2 章 视频编码标准和视频格式 …… 31

2.1 前言 …… 31
2.2 复杂度的降低 …… 33
2.3 视频编码标准 …… 33
2.4 MPEG 和 H.26x …… 34
2.4.1 H.120 …… 34
2.4.2 H.261 …… 34
2.4.3 MPEG-1 …… 35
2.4.4 H.262/MPEG-2 …… 35
2.4.5 H.263，H.263+和 H.263++ …… 35
2.4.6 MPEG-4 …… 36
2.4.7 H.264/MPEG-4 第 10 部分/AVC …… 36
2.4.8 H.265/HEVC …… 37
2.5 视频格式和质量 …… 37
2.5.1 帧与场 …… 38
2.5.2 颜色空间 …… 38

2.5.3 视频格式 …… 40
2.5.4 质量 …… 41
2.6 总结 …… 42

第3章 AVS China …… 43

3.1 AVS China …… 43
3.2 AVS China 档次和级别 …… 45
3.2.1 AVS-video 的基准档次 …… 46
3.2.2 AVS-video 基本档次 …… 46
3.2.3 AVS-video 伸展档次 …… 46
3.2.4 AVS-video 加强档次 …… 46
3.3 AVS 使用的数据格式 …… 47
3.3.1 AVS 视频分层结构 …… 47
3.4 AVS 视频编码器 …… 49
3.4.1 编码过程概述 …… 50
3.4.2 AVS 视频编码器用到的编码工具 …… 50
3.5 AVS 视频解码器 …… 57
3.6 AVS 视频比特流 …… 57
3.6.1 起始码 …… 57
3.6.2 起始码值 …… 58
3.6.3 帧编码类型 …… 59
3.7 AVS 视频流的 NAL 单元 …… 59
3.7.1 AVS 视频流 NAL 单元的映射 …… 59
3.7.2 NAL 单元首部描述 …… 60
3.8 AVS-M 简介（AVS 第 7 部分） …… 61
3.8.1 AVS-M 的数据结构 …… 62
3.8.2 AVS-M 的体现 …… 65
3.8.3 基本档次的多个级别 …… 66
3.9 块模式预测模式 …… 67
3.9.1 帧内预测 …… 67
3.9.2 帧间预测 …… 69
3.9.3 跳跃模式预测 …… 70
3.9.4 率失真优化 …… 71
3.10 变换、量化和熵编码 …… 71
3.10.1 变换 …… 71
3.10.2 量化 …… 71

3.10.3 熵编码 …… 72
3.10.4 简化的去块效应滤波器 …… 72
3.11 AVS 第 1 部分：系统 …… 74
3.11.1 程序流 …… 75
3.11.2 运输流 …… 78
3.12 IEEE AVS …… 80
3.12.1 应用 …… 81
3.12.2 档次和级别 …… 81
3.12.3 设计特征概述 …… 81
3.13 总结 …… 81
3.14 专题研究 …… 82

第 4 章 H.264/MPEG-4 先进视频编码 …… 83

4.1 概述 …… 83
4.2 H.264 的档次和级别 …… 84
4.2.1 H.264 的档次 …… 84
4.2.2 H.264 的级别 …… 88
4.3 H.264 编码器 …… 88
4.4 帧内预测 …… 89
4.5 帧间预测 …… 91
4.6 P 分片中宏块的帧间预测 …… 91
4.7 亚像素运动矢量 …… 92
4.8 变换与量化 …… 94
4.9 环路去块效应滤波器 …… 95
4.9.1 滤波器强度 …… 96
4.10 B 分片与自适应加权预测 …… 97
4.11 熵编码 …… 98
4.12 H.264 解码器 …… 100
4.13 H.264 的一些应用 …… 100
4.14 总结 …… 101
4.15 专题研究 …… 102

第 5 章 高效视频编码（HEVC） …… 105

5.1 引言 …… 105
5.2 视频编码联合协作团队 …… 105
5.3 HEVC 测试模型中编码工具的分析，HM 1.0：帧内预测 …… 111

5.4　HEVC 编码器 …… 112
5.4.1　帧内预测 …… 114
5.4.2　变换系数扫描 …… 117
5.4.3　亮度与色度的分数像素内插 …… 118
5.4.4　HM1 与 HEVC 草案 9 编码工具的比较 …… 119
5.5　扩展到 HEVC …… 119
5.6　档次和级别 …… 121
5.7　HEVC 编码器的性能和计算复杂度 …… 122
5.8　HEVC 的系统层面集成 …… 122
5.9　HEVC 的无损编码与改进 …… 123
5.10　总结 …… 124
5.11　专题研究 …… 125

第 6 章　VP6 视频编码标准 …… 136

6.1　前言 …… 136
6.2　与以前的 Flash 编译码系统 MX 相比 …… 136
6.3　VP6 算法基础 …… 141
6.4　VP6 的编码档次 …… 143
6.5　帧类型 …… 143
6.5.1　黄金帧 …… 144
6.6　宏块模式 …… 145
6.6.1　I 帧的宏块模式（帧内模式） …… 145
6.6.2　P 帧的宏块模式（帧间模式与帧内模式） …… 145
6.7　最近邻块和近邻块 …… 146
6.8　运动矢量 …… 147
6.8.1　编码 …… 147
6.8.2　预测环路滤波 …… 148
6.9　分数像素运动补偿的滤波 …… 148
6.9.1　双线性滤波 …… 148
6.9.2　双三次滤波 …… 149
6.10　对于无约束运动矢量的支持 …… 153
6.11　预测环路滤波 …… 154
6.12　DCT，扫描顺序和系数牌集合 …… 155
6.12.1　扫描顺序 …… 155
6.12.2　DCT 编码与系数牌集合 …… 157
6.13　量化 …… 161

6.14 熵编码 …… 162
6.14.1 上下文信息的利用 …… 163
6.14.2 霍夫曼编码器 …… 163
6.14.3 BoolCoder …… 163
6.15 VP6 编码概述 …… 164
6.16 VP6 的编码性能 …… 164
6.17 VP6 黄金帧 …… 165
6.18 背景/前景分割 …… 166
6.19 上下文预测的熵编码 …… 167
6.20 比特流分割 …… 167
6.21 双模式算术与变长编码 …… 168
6.22 自适应亚像素运动估计 …… 168
6.23 VP6-E 和 VP6-S 编码器档次 …… 168
6.24 设备端口与硬件实现 …… 169
6.25 总结 …… 170
6.26 专题研究 …… 171

第 7 章 Dirac 视频编译码器与 H.264/MPEG-4 第 10 部分的性能分析和对比 …… 172

7.1 前言 …… 172
7.2 Dirac 的体系结构 …… 173
7.2.1 Dirac 编码器 …… 173
7.2.2 Dirac 解码器 …… 174
7.3 Dirac 中的编码和解码阶段 …… 174
7.3.1 小波变换 …… 174
7.3.2 缩放和量化 …… 176
7.3.3 熵编码 …… 177
7.3.4 运动估计 …… 177
7.3.5 运动补偿 …… 179
7.3.6 解码器 …… 180
7.4 实现 …… 180
7.4.1 编码结构概述 …… 181
7.4.2 编码的简明性和相对速度 …… 181
7.5 结果 …… 182
7.5.1 压缩比测试 …… 182
7.5.2 SSIM 测试 …… 184

7.5.3　PSNR 测试 …… 185
7.6　结论 …… 188
7.7　未来的研究 …… 189
7.8　总结 …… 189
7.9　专题研究 …… 189

第 8 章　VC-1 视频编码 …… 191

8.1　VC-1 的结构 …… 191
8.2　整数变换编码 …… 192
8.2.1　逆变换 …… 192
8.2.2　前向变换 …… 193
8.3　运动估计/补偿 …… 195
8.3.1　环路滤波器 …… 196
8.3.2　复杂度 …… 197
8.3.3　档次和级别 …… 197
8.4　简单档次 …… 200
8.4.1　比特流结构 …… 200
8.4.2　基本档次的帧内压缩 …… 201
8.4.3　块大小可变的变换规定 …… 202
8.4.4　重叠变换 …… 204
8.4.5　每个宏块的 4 个运动矢量 …… 206
8.4.6　Y 分量的 1/4 像素运动补偿 …… 207
8.5　主档次 …… 211
8.5.1　CbCr 的 1/4 像素运动补偿 …… 211
8.5.2　起始码 …… 211
8.5.3　扩展的运动矢量 …… 212
8.5.4　环路滤波器 …… 213
8.5.5　动态分辨率的改变 …… 215
8.5.6　B 帧 …… 215
8.5.7　自适应宏块量化 …… 216
8.5.8　亮度补偿 …… 218
8.5.9　范围调节 …… 219
8.6　高级档次 …… 220
8.6.1　比特流结构 …… 220
8.6.2　隔行扫描 …… 220
8.6.3　序列级的用户数据 …… 222

8.6.4 入口点层 …… 223
8.6.5 显示元数据 …… 223
8.7 H.264 到 VC-1 的转码 …… 224
8.7.1 帧内编码宏块模式映射 …… 225
8.7.2 帧间编码宏块模式映射方案 …… 226
8.7.3 运动矢量映射 …… 227
8.7.4 参考帧 …… 227
8.7.5 跳过宏块 …… 227
8.8 VC-1 的传输 …… 227
8.8.1 传输流中 VC-1 的数据封装 …… 229
8.8.2 程序流中的 VC-1 数据封装 …… 230
8.9 VC-2 视频压缩 …… 231
8.9.1 前言 …… 232
8.9.2 范围 …… 233
8.10 总结 …… 233
8.11 专题研究 …… 233

附录 A 关于 Dirac、H.264 和 H.265 图像质量的调查研究 …… 235

A.1 前言 …… 235
A.2 H.265 …… 235
A.3 使用 SSIM 和 FSIM 的图像质量评价 …… 236
A.4 结果 …… 241
A.4.1 使用 QCIF 序列 Foreman 的结果 …… 241
A.4.2 使用 CIF 序列 Foreman 的结果 …… 245
A.4.3 使用 QCIF 序列 Container 的结果 …… 245
A.4.4 使用 CIF 序列 Container 的结果 …… 252
A.5 结论 …… 256
A.6 专题研究 …… 257

附录 B AVSNR 软件的 PSNR 平均值 …… 258

参考文献 …… 258

附录 C 通用图像质量指标与 SSIM 的比较 …… 259

C.1 前言 …… 259
C.1.1 MSE： …… 259
C.1.2 为什么用 MSE？［Q22］ …… 260

C. 1. 3　MSE 有什么问题？［Q22］ …… 260
C. 1. 4　使用 MSE 的隐含假定 …… 262
C. 1. 5　主观对客观图像质量测度 …… 262
C. 2　通用的图像质量指标 …… 263
C. 2. 1　质量指标的定义 …… 263
C. 2. 2　在图像中的应用 …… 264
C. 3　结构相似度指标 …… 266
C. 4　带有失真的图像 …… 270
C. 5　测试结果 …… 274
C. 6　结论 …… 276
专题研究 C　图像知觉质量评价指标：结构相似度 …… 277

附录 D　H. 264 中模式相关的 DCT/DST 实现方案 …… 279

D. 1　前言 …… 279
D. 2　参考软件中变换的实现方案 …… 280
D. 3　提出的方案 …… 281
D. 3. 1　从帧内预测模式到 DCT/DST 的映射 …… 281
D. 3. 2　获得 H. 264 的 DST 矩阵 …… 281
D. 3. 3　H. 264/AVC 参考软件中 DCT/DST 的实现 …… 283
D. 4　BD-PSNR 和 BD-比特率的计算 …… 284
D. 5　性能分析 …… 284
D. 5. 1　WQVGA（416×240）序列的测试结果 …… 284
D. 5. 2　WVGA（832×480）序列的测试结果 …… 287
D. 5. 3　HD（1920×1080）序列的测试结果 …… 290
D. 5. 4　高清（1080×720）序列的测试结果 …… 293
D. 5. 5　用于 RaceHorse 序列 DCT/DST 不同组合的测试结果 …… 296
D. 6　结论与未来的工作 …… 298

附录 E　H. 264 的实现软件 JM，Intel IPP 和 x264 的性能分析与比较 …… 300

E. 1　H. 264 …… 300
E. 2　JM 软件 …… 302
E. 3　x264 …… 302
E. 4　Intel IPP …… 302
E. 5　JM（17. 2）性能分析 …… 304
E. 6　x264 性能分析 …… 307
E. 7　Intel IPP 性能分析 …… 310

E.8 在基本档次、主档次和高档次中 JM、x264 和 Intel IPP 软件的 SSIM 比较 …… 313
E.9 基本档次、主档次和高档次中 JM、x264 和 Intel IPP 软件实现的 PSNR 对比 …… 315
E.10 基本档次、主档次和高档次中 JM、x264 和 Intel IPP 软件所用编码时间的对比 …… 317
E.11 基本档次、主档次和高档次中 JM、x264 和 Intel IPP 软件实现的压缩比对比 …… 319
E.12 结论 …… 321
E.13 未来的工作 …… 321

附录 F 基于 H.264“只编码 I 帧”的 AIC 实现以及与其他静止帧图像编码标准如 JPEG、JPEG 2000、JPEG-LS 和 JPEG-XR 的比较 …… 322

F.1 前言 …… 322
F.2 高级图像编码 …… 323
F.3 改进的 AIC …… 327
F.4 H.264 标准 …… 329
F.4.1 H.264/AVC 主档次帧内编码 …… 330
F.4.2 H.264/AVC FRExt 高档次帧内编码 …… 330
F.5 JPEG …… 331
F.6 JPEG2000 …… 333
F.7 JPEG XR …… 334
F.8 JPEG-LS …… 335
F.9 JPEG-LS 算法 …… 336
F.10 主要差别 …… 338
F.11 评价方法 …… 339
F.11.1 图像测试序列 …… 339
F.11.2 编译码器的设置 …… 339
F.11.3 主观对客观图像质量测度 …… 341
F.12 结论和未来的工作 …… 342

附录 G 面向高清视频编码的更高阶二维整数余弦变换 …… 371

G.1 离散余弦变换与视频压缩 …… 371
G.2 整数余弦变换 …… 372
G.3 简单的 2 维 16 阶 ICT …… 374
G.4 改进的 2 维 16 阶 ICT …… 378

G. 5　基于 Loeffler 因式分解的二维 16 阶 binDCT …… 381
G. 6　变换编码增益 …… 384
G. 7　H. 264/AVC 中的实现方案与性能分析 …… 386
G. 8　AVS-Video 的实现方案和性能分析 …… 388
G. 9　结论和未来的工作 …… 390

参考文献 …… 392

缩　略　语

2D-VLC　Two-Dimensional Variable Length Coding　二维变长编码
AAC　Advanced Audio Coder　高级音频编码器
AAD　Adaptive Arithmetic Decoder　自适应算术解码器
ACM　Association for Computing Machinery　美国计算机协会
ACR　Absolute Category Rating　绝对分类评级法
ADPCM　Adaptive DPCM　自适应 DPCM
AES　Audio Engineering Society　音频工程学会
AIC　Advanced Image Coding　高级图像编码
AIF　Adaptive Interpolation Filter　自适应内插滤波器
ALF　Adaptive Loop Filter　自适应环路滤波器
AMVP　Adaptive Motion Vector Prediction　自适应运动矢量预测
ANSI　American National Standards Institute　美国国家标准局
APIF　Adaptive Pre-Interpolation Filter　自适应预内插滤波器
APSIPA　Asia Pacific Signal and Image Processing Association　亚太信号与图像处理协会
ARM　Advanced RISC Microprocessor　高级 RISC 微处理器
ASIC　Application Specific Integrated Circuits　专用集成电路
ASO　Arbitrary Slice Order　任意分片顺序
ASSP　Acoustics, Speech and Signal Processing　声学、语音和信号处理
ATSC　Advanced Television System Committee　美国高级电视系统委员会
AU　Access Unit　访问单元
AVC　Advanced Video Coding　先进视频编码
AVS　Audio and Video coding Standard of China　中国音视频编码标准
AVS-M　Audio Video Standard for Mobile　移动音视频标准
AWGN　Additive White Gaussian Noise　加性高斯白噪声
Bslice　Bi-directionally predictive Slice　双向预测片
BBC　British Broadcasting Corporation　英国广播公司
BD　Bjøntegaard Δ（译者注：用来比较两种编码方法的性能指标）
B-Frame　Interpolated Frame　B 帧（译者注：英文全称疑似 Bidirectional Interpolated Frame）
BMP　Bit Map format　位图格式

bps	Bits per second 比特每秒（译者注：本书中采用 bit/s 的表示方法）
BSTM	Butterfly Style Transform Matrices 蝶形变换矩阵
BT	Broadcasting service（Television） 广播业务（电视）
C2DVLC	The Context-based 2D Variable Length Coding 基于上下文的二维变长编码
CABAC	Context-based Adaptive Binary Arithmetic Ccoding 基于上下文的自适应二进制算术编码
CAFI	Combined Adaptive-Fixed Interpolation with multi-directional filters 结合多向滤波器的自适应-固定内插法
CAVLC	Context-Adaptive Variable Length Coding 上下文自适应变长编码
CB	Coding Block 编码块
CBAC	Context-based Binary Arithmetic Coding 基于上下文的二进制算术编码
CBP	Coded Block Pattern 编码块模式
CBR	Constant Bit Rate 恒定比特率
CC	Color Conversion 颜色转换
CCD	Charge Coupled Devices 电荷耦合器件
CD	Committee Draft 委员会草案
CE	Core Experiment 核心实验
CI	Confidence Interval 置信区间
CIF	Common Intermediate Format 公用中介格式
CODEC	Coder and Decoder 编解码器
CRA	Clean Random Access 完全随机访问
CSIQ	Categorical Subjective Image Quality 按类别的主观图像质量
CSNR	Channel Signal to Noise Ratio 信道信噪比
CSVT	Circuits and Systems for Video Technology 视频技术的电路与系统
CTB	Coding Tree Block 码树块
CTU	Coding Tree Unit 码树单元
CU	Coding Unit 编码单元
dB	Decibel 分贝
DCC	Data Compression Conference 数据压缩会议
DCT	Discrete Cosine Transform 离散余弦变换
DDCT	Directional DCT 有向 DCT
DIP	Direct Intra Prediction，Digital Image Processing 直接帧内预测，数字图像处理
DMB	Digital Multimedia Broadcasting 数字多媒体广播

DMOS　Differential Mean Opinion Score　差异平均观点得分
DPB　Decoded Picture Buffer　解码图像缓存
DPCM　Differential Pulse Code Modulation　差分脉冲编码调制
DRM　Digital Rights Management　数字版权管理
DSCQS　Double Stimulus Continuous Quality Scale　双激励连续质量评级法
DSIS　Double Stimulus Impairment Scale　双激励损伤评级法
DSP　Digital Signal Processing　数字信号处理
DTS　Decoding Time Stamp　解码时间标记
DVB　Digital Video Broadcasting　数字视频广播
DVD　Digital Versatile Disk　数字通用光盘
DWT　Discrete Wavelet Transform　离散小波变换
EBCOT　Embedded Block Coding with Optimized Truncation　优化截取嵌入式块编码
EBU　European Broadcasting Union　欧洲广播联盟
EE　Electrical Engineering　电气工程
EEQM　Energy Equalization Quality Metric　能量均衡质量指标
EI　Electronic Imaging　电子成像
EOB　End Of Block　块尾
ES　Elementary Stream　基本码流
EZW　Embedded Zero-tree Wavelet coding　嵌入式零树小波编码
FAAC　Free Advanced Audio Coder　免费高级音频编码器
FDCT　Forward DCT　前向 DCT
FDIS　Final draft international standard　国际标准最终草案
FFT　Fast Fourier Transform　快速傅里叶变换
FLAC　Free Lossless Audio Codec　免费无损音频编解码器
FMO　Flexible Macro Block Order　灵活的宏块顺序
fps　Frames per second　帧每秒（译者注：本书采用 f/s 的表示方法）
FR　Frame Rate，Full Refetence　帧率，全参考
FRExt　Fidelity Range Extensions　保真度范围扩展
FSIM　Feature Similarity Index　特征相似度指数
GIF　Graphics Interchange Format　图形交互格式
GM　Gradient Magnitude　梯度幅值
GOP　Group Of Picture（s）　帧组
HD　High Definition　高清晰度
HD-photo　High-Definition photo　高清照片
HDTV　High Definition Television　高清电视

HE　High Efficiency　高效率
HEVC　High Efficiency Video Coding　高效视频编码
HHR　Half-Horizontal Resolution, Horizontal High Resolution　半水平分辨率，水平高分辨率
HM　HEVC test Model　HEVC 测试模型
HP　High Profile　高级档次
HR　High Resolution　高分辨率
HVR　Half-Vertical Resolution　半垂直分辨率
HVS　Human Visual System　人类视觉系统
Islice　Intra slice　内部片
IASTED　International Association of Science and Technology for Development　国际科技发展协会
ICALIP　International Conference o Audio, Language and Image Processing　音频、语言和图像处理国际会议
ICC　Inverse CC　逆颜色变换
ICIP　International Conference on Image Processing　图像处理国际会议
ICT　Integer Cosine Transform　整数余弦变换
IDR　Instantaneous Decoding Refresh　瞬时解码刷新
IEC　International Electrotechnical Commission　国际电工委员会
IEEE　Institute of Electrical and Electronics Engineers　电气电子工程师协会
I-Frame　Intra Frame　内部编码帧
IJWMIP　International Journal on Wavelets, Multimedia and Image Processing　小波、多媒体和图像处理国际期刊
ILF　In-Loop Filtering　环内滤波
IMS　IP Multimedia Subsystem　IP 多媒体子系统
IPR　Intellectual Property Rights　知识产权
IPTV　Internet Protocol TV　交互式网络电视
IS　International Standard　国际标准
ISCAS　International Symposium on Circuits and Systems　电路与系统国际研讨会
ISDB　Integrated Services Digital Broadcasting　综合业务数字广播
ISO　International Standardization Organization　国际标准化组织
IST　Integer Sine Transform　整数正弦变换
ITU-T　The Telecommunication Standardization Sector of the International Telecommunication Union　国际电信联盟电信标准化部门
IVC　Imagesand Video Communications　图像与视频通信
IZZ　Inverse Zig-Zag scan　逆 Z 形扫描

JAES	Journal of AES　AES 期刊
JM	Joint Model　联合模型
JPEG	Joint Photographic Experts Group　联合图像专家组
JPEG-LS	JPEG Lossless Coding　JPEG 无损编码
JPEG-XR	JPEG Extended Range　JPEG 延伸范围
JVCIR	Journal of Visual Communication and Image Representation　视觉通信与图像表示期刊
JVT	Joint Video Team　联合视频小组
kbps	kilo bps　千比特每秒（译者注：本书采用 kbit/s 的表示方法）
KLT	Karhunen-Loeve transform　KL 变换
KTA	Key Technical Areas　关键技术领域
LAN	Local Area Network　局域网
LBT	Lapped Bi-orthogonal Transform　重叠双正交变换
LIVE	Laboratory for Image and Video Engineering　图像与视频工程实验室
LOCO-I	Low Complexity Lossless Compression for Images　低复杂度无损图像压缩
LR	Low Resolution　低分辨率
LZW	Lempel-Ziv-Welch　LZW 编码（译者注：是一种“以兰佩-齐夫-韦尔奇：三个人名命名的字典编码方法）
M-AIC	Modified Advanced Image Coding　改进的高级图像编码
MB	Macroblock　宏块
MBAFF	Macroblock-level Adaptive Frame/Field coding　宏块级自适应帧/域编码
Mbps	Megabps　兆比特每秒（译者注：本书采用 Mbit/s 的表示方法）
MC	Motion Compensation　运动补偿
MCPE	Motion Compensated Prediction Error　运动补偿预测误差
MDCT	Modified DCT　改进的 DCT
MDDT	Mode Dependent Directional Transform　模式相关的方向变换
ME	Motion Estimation　运动估计
MICT	Modified ICT　改进的整数余弦变换
MICTL	Media Information and Communication Technology Laboratory　媒体信息与通信技术实验室
MMS	Multimedia Messaging Service　多媒体信息业务
MMSE	Minimum MSE，Multimedia Signal Processing　最小均方误差，多媒体信号处理
MOMS	Maximal-Order Interpolation with Minimal Support　最小支撑的最大阶内插

MOS	Mean Opinions Core	平均意见得分
MOVs	Mean Outpu Tvariables	输出变量均值
MP	Main Profile	主档次
MPEG	Moving Picture Experts Group	运动图像专家组
MPM	Most Probable Mode	最可能模式
MSSSIM	Multi Scale Structural Similarity	多尺度结构相似度
MSE	Means Quare Error	均方误差
MSSIM	Mean Structural Similarity	平均结构相似度
MSU	Moscow State University	莫斯科国立大学
MUSHRA	Multiple Stimulus with Hidden Reference and Anchors	隐含参考和锚的多激励法
MV	Motion Vector	运动矢量
NAB	National Association of Broadcasters	全美广播电视协会
NAL	Network Abstraction Layer	网络抽象层
NMSE	Normalized Mean Square Error	归一化的均方误差
NO	No Reference	无参考
NORM	No-reference Video Quality Monitoring Algorithm	无参考视频质量监视算法
NTSC	National Television System Committee	国家电视制式委员会
OBMC	Overlapped Block-based Motion Compensation	重叠块运动补偿
ODG	Overall Difference Grade	整体差分等级
OLT	Over Lapped Transform	重叠变换
P slice	Predictive slice	预测片
PAFF	Picture-level Adaptive Frame/Field coding	图像级自适应帧/场编码
PAL	Phase Alternating Line	逐行倒相制式
PC	Phase Congruency	相位一致性
PCM	Pulse Code Modulation，Pacific-Rim Conference on Multimedia	脉冲编码调制，环太平洋多媒体会议
PCS	Picture Coding Symposium	图像编码研讨会
PDMOSCL	Predicted Degradation of Differential Mean Opinion Score due to Coding-artifacts and Packet Losses	编码损伤和丢包引起的差分意见得分均值的预测退化
PE	Prediction Error	预测误差
PEAQ	Perceptual Evaluation of Audio Quality	音频质量感知评价
PES	Packetized Elementary Stream	包化基本流
P-Frame	Predicted Frame	预测帧

PGM	Portable Gray Map	便携式灰度图（译者注：一种图像格式）
PID	Packet Identifier	包标识
PIT	Prescaled Integer Transform	预缩放整数变换
PMT	Program Map Table	程序图表
PNM	Portable Any Map	便携式任意图（译者注：图像格式）
PPM	Portable Pixel Map	便携式像素图（译者注：图像格式）
PPS	Picture Parameter Set	图像参数集
PS	Program Stream	程序流
PSI	Program Specific Information	程序特定信息
PSNR	Peak-to-peak Signal-to-Noise Ratio	峰值信噪比
PU	Prediction Unit	预测单元
PVR	Personal Video Recorder	个人视频记录仪
Q	Quantization	量化
QCIF	Quarter CIF	四分之一 CIF（译者注：大小为 1/4CIF 的图像格式）
QF	Quality Factor	质量因数
QM	Quality Metric	质量测度
QMF	Quadrature Mirror Filter	求积镜像滤波器
QP	Quantization Parameter	量化参数
RAM	Random Access Memory	随机存取存储器
R-QF	Rate QF	率-质量因数（函数关系-译者注）
RDCost	Rate Distortion Cost	率失真代价
RDO	Rate Distortion Optimization	率失真优化
RGB	Red, Green and Blue	红绿蓝（译者注：一种颜色空间）
RL	Run-Length	游程
RMS	Root Mean Square	方均根值
RMSE	Root Mean Square Error	方均根误差
ROI	Region of Interest	感兴趣区域
ROT	Rotational Transform	旋转变换
RR	Reduced Reference	减少的参考（帧）
RV	Real Video	真实视频
RVC	Reconfigurable Video Coding	可重构视频编码
SAD	Sum of Absolute Differences	绝对误差和
SATD	Sum of Absolute Transform Differences	绝对变换误差和
SBR	Spectral Band Replication	谱带重现
SD	Standard Definition	标准清晰度
SDO	Standards Development Organizations	标准发展组织

SEI	Supplemental Enhancement Information 增补的增强信息
SI	Switched Intra 切换的内部编码模式
SICT	Simple Order 16 ICT 简单 16 阶整数余弦变换
SIF	Source Input Format 源输入格式
SIMD	Single Instruction Multiple Data 单指令多数据
SMPTE	Society of Motion Picture and Television Engineers 电影电视工程师协会
SNR	Signal to Noise Ratio 信噪比
SoC	System on Chip 片上系统
SP	Switch Ingprediction, Switched Predictive 切换预测
SPS	Sequence Parameter Set 序列参数集
SSCQE	Single Stimulus Continuous Quality Evaluation 单激励连续质量评法
SSE	Streaming SIMD Extensions 流 SIMD 扩展
SSIM	Structural Similarity 结构相似度
SVC	Scalable Video Coding 可伸缩视频编码
SSVC	Spatially SVC 空域可伸缩编码
TB	Transform Block 变换块
TE	Tool Experiments 工具实验
TF-MSSIM	Time-Frequency MSSIM 视频 MSSIM
TID2008	Tampere Image Database 2008 坦佩雷图像数据库 2008
T-MSSIM	Temporal MSSIM 时域 MSSIM
TS	Transport Stream 运输流
T-STD	Transport System Target Decoder 运输系统目标解码器
TU	Transform Unit 变换单元
UIQI	Universal Image Quality Index 通用图像质量指标
UTA	University of Texas at Arlington 得克萨斯大学阿灵顿分校
UVLC	Universal Variable Length Coding 通用变长编码
VBR	Variable Bit Rate 可变比特率
VC-1	Video Coder 1 by SMPTE SMPTE 视频编码器 1 号
VCEG	Video Coding Experts Group 视频编码专家组
VCIP	Visual Communications and Image Processing 视觉通信与图像处理
VCIR	Visual Communications and Image Representation 视觉通信与图像表示
VCL	Video Coding Layer 视频编码层
VLC	Variable Length Coding, variable length codes 变长编码，变长码
VLD	Variable Length Decoder 变长解码器

VLSI　　Very Large Scale Integration　超大规模集成（电路）
VOD　　Video on Demand　视频点播
VQ　　Vector Quantization　矢量量化
VQEG FR-TV　　Video Quality Experts Group Full Reference Television　视频质量专家组全参考电视
VQM　　Video Quality Model，Video Quality Metric　视频质量模型，视频质量测度
WD　　Working Draft　工作草案
WMA　　（Microsoft）Windows media audio　（微软）Windows 媒体音频（译者注：一种音频格式）
WQVGA　　Wide Quarter Video Graphics Array　宽屏 1/4 视频图形阵列（译者注：一种显示屏分辨率）
WVGA　　Wide Video Graphics Array　宽屏视频图形阵列
YUV　　Luminance and Chrominance Color Components　亮度和色度颜色分量
ZZ　　Zig-Zag scan　Z 形扫描

第 1 章　引　　言

摘要：除简述了各种视频/图像编码标准，如 JPEG、MPEG 和 H.26x 系列之外，还介绍了 DIRAC（第 7 章），AVS China（第 3 章）和 VC-1（第 8 章）等标准。类似地简述了音频编码之后，以指标对比总结本章。

关键词：标准　视频编译码　音频编译码　颜色空间　主观质量　客观质量测度　音频质量　视频质量　性能

1.1　流行的视频和音频标准

多个来源的视频、音频、图像和文本的组合体通常称为多媒体。过去二十年里，在电视、娱乐、互动业务、电信、会议、互联网、消费电子设备、医疗、安全、商业、交通、国防和银行等广泛的领域里，对多媒体通信的要求迅速增长。通常，因为原始数据量太大，音频和视频数据在存储或者传输之前必须要进行压缩。而且在显示或进一步处理之前压缩数据应该能被解码。压缩也被称为编码，而解压则被称为解码。因此，压缩和解压视频/音频数据的软件或者硬件设备分别叫作视频/音频编码器和解码器。方便起见，编码器和解码器缩写为 codec。尽管已经开发出了许多视频和音频编码算法，但实现多媒体通信的却是视频和音频编码标准，这些标准能保证多个商家提供的软件和硬件具有互操作性。视频和音频标准系列已经由标准发展组织（SDO）和著名公司开发出来，这些组织包括 ISO/IEC（国际标准化组织和国际电工委员会）[H53] [H54]、ITU-T（国际电信联盟的电信标准化部门，以前的 CCITT）[H55]、SMPTE（运动图像与电视工程师协会）[C31]，AVS China（中国的音频与视频编码标准）[AS1]、BBC 的 DIRAC [D1] [D5]；这些公司包括微软 [C32]、Real Networks [R3] 和 On2 Technology [P11]。

ISO/IEC 已经开发了几个视频和音频标准，包括 MPEG-1（ISO/IEC 11172）[S2]，MPEG-2（ISO/IEC 13818）[S3]，和 MPEG-4（ISO/IEC 14496）[S8]。ITU-T 也已经开发了若干标准，但不像 MPEG 那样，视频标准和音频标准是分开的。ITU-T H.261 [S4]，H.262 [S3]㊀，H.263 [S5]，H.263+(H.263 第 2 版)[S6] [S7]，H.26L [PC1] 和 H.264 [S10] [H23] 都是为视频设计的。而 ITU-T G.723.1 [O14] 和 G.729 [O15] 则是面向音频的。除此之外，还有 SMPTE 开发的

㊀ H.262 在不少部分中也有音频编码。

视频编码器 1 号（VC-1）[C11] [C14] 和视频编码器 2 号（VC-2）[C6]，微软的 Windows 媒体视频 9（Windows Media Video 9），On 2 Technology 公司的 VP6 [P3] [P5] [P6] 和 VP7 [P4]，BBC 的 Dirac，Real Networks 公司的 Real Video 9 和 Real Video 10 [R1] [R2] 也都是流行于互联网和个人计算机上的视频标准。近年来，AVS China [A2，A10，A59-A66] 吸引了全世界与电视、多媒体通信、甚至芯片制造相关的行业的大量关注。这个新标准包括四个主要的技术领域，分别是系统、视频、音频和数字版权管理，以及一些支持文档如一致性验证。该标准的第二部分被称为 AVS1-P2 [A2]（视频-基准）于 2006 年获批成为中国国家标准。该标准的几个最终草案已经完成，包括 AVS1-P1（系统）[A1]，AVS1-P2（视频-增强）[A3]，AVS1-P3（音频）[A4] 和 AVS1-P7（移动视频）[A74]。

AVS China 提供了优化的编码性能并具有最低的总体成本，包括传输和存储成本、实现成本和知识产权成本，因为 AVS China 工作组从建立标准之初就考虑技术知识产权成本的问题。知识产权成本有两个方面。一是内容的知识产权，这不在音频和视频编码标准之外内。另一方面是音频和视频编码标准所用到的技术的知识产权成本。

有关 AVS China 最近的一些研究成果报道于“信号处理：图像通信”期刊专辑上的几篇论文 [A58-A66]。[A59] 给出了 AVS 视频部分的概述，也描述了可用的编码工具，并给出 AVS 定义的应用驱动的档次的例子。[A60] 介绍了 AVS 视频编码标准的两个基于上下文的熵编码方案。一个是基于上下文的 2D 变长编码（C2DVLC），作为 AVS 第二部分基准档次的低复杂度熵编码方案。另一个是基于上下文的二进制算术编码（CBAC），作为 AVS 第二部分增强档次的增强熵编码方案。[A61] 提出了一种亚像素内插滤波器，被称为结合多向滤波器的自适应-固定内插法，能以低计算复杂度来获得良好的编码效率。此外，该专辑也讨论了算法实现 [A64] [A65]、可重构视频编码（RVC）框架 [62]，技巧模式 [A63] 和鲁棒双水印算法 [A66]。

这些流行的视频标准列表见表 1.1，这些标准的算法元素对比见表 1.2。表 1.1 选自 T. Ebrahimi 和 M. Kunt 的论文“Visual data compression for multimedia applications”，Proc. IEEE，第 86 卷：1109-1125，1998 年 6 月 [G1]，不过做了些微小的改动。最近的标准诸如 H.264/MPEG-4 第 10 部分，Dirac，AVS China，JPEG-LS，JPEG-XR，JBIG，VC-1（SMPTE），VC-2，HEVC/NGVC 和 VP6（目前 VC10）也加在本书当中。

表 1.1　流行的视频标准 [G1]（IEEE 1998）

标　　准	标准化团体	主要目标比特率	主要压缩技术	主要目标应用
JPEG	ISO/IEC	压缩比2～30	DCT 视觉量化 Z 形重排序 霍夫曼编码 算术编码	互联网成像 数字摄影 图像与视频编辑

（续）

标 准	标准化团体	主要目标比特率	主要压缩技术	主要目标应用
JPEG2000	ISO/IEC	压缩比 2～50	小波 EBCOT	互联网成像 数字摄影 图像与视频编辑 印刷 医学成像 移动应用 颜色传真 卫星成像 感兴趣区域编码，图片与艺术品数字化文库，按像素精度或分辨率的渐进传输，激光打印的渲染，印前影像
JPEG-LS 第 1 部分与第 2 部分	ISO/IEC		第 1 部分 上下文建模 预测 Golomb 码	连续色调静止图像的无损与近无损编码
			第 2 部分 算术编码 更有效的预测 改进的 Golomb 码	连续色调静止图像的无损与近无损编码
JBIG1，JBIG2	ISO/IEC JTC/SC 29/WG1 ITU-T	JBIG1：压缩比 20:1	JBIG1 基于上下文的算术编码器（动态适应于每个像素内容的统计规律） 渐进/序贯编码	传真机的渐进（有损到无损）编码
		JBIG2：压缩比高于 JBIG1 的 2～4 倍	JBIG2 有损、无损以及有损到无损的图像压缩 高质量渐进编码 内容渐进编码 基于内容的（文本、半色调、线条稿、大字符）分解和编码 基于内容的编码、软模式匹配、模式匹配、和置换 二值图像编码后封装的文件格式 数据（顺序访问和随机访问）	公文存储与存档 万维网图像编码 无线数据传输 假脱机打印 远程会议 互动多媒体

（续）

标　　准	标准化团体	主要目标比特率	主要压缩技术	主要目标应用
JPEG-XR（AIC）	ISO/IEC		基于微软的高清图片（Windows 媒体图片）	连续色调摄影内容的存储与交换（有损和无损）
MPEG-1	ISO/IEC	最高达 1.5Mbit/s	DCT 视觉量化 自适应量化 Z 形重排序 预测运动补偿 双向运动补偿 半像素精度运动估计 霍夫曼编码 算术编码	CD-ROM 存储 消费者视频 数码相机
MPEG-2	ISO/IEC	从 1.5Mbit/s 直到 大约 3.5Mbit/s	DCT 视觉量化 自适应量化 Z 形重排序 预测运动补偿 双向运动补偿 基于帧/场的运动补偿 半像素精度运动估计 空域可伸缩性 质量可伸缩性 霍夫曼编码 算术编码 容错编码	数字电视 数字高清电视 高质量视频 卫星电视 有线电视 地面广播 视频编辑 视频存储 数码相机
MPEG-4 第 2 部分视觉	ISO/IEC	8kbit/s ~ 大约 35Mbit/s 的比特率	DCT 小波 视觉量化 自适应量化 Z 形重排序 预测运动补偿 双向运动补偿 基于帧/场的运动补偿 半像素精度运动估计 高级运动估计 重叠运动补偿	互联网 互动视频 视觉编辑 内容操作 消费者视频 专业视频 2D/3D 计算机图形

（续）

标　　准	标准化团体	主要目标比特率	主要压缩技术	主要目标应用
			空域可扩展性 时域可扩展性 质量可扩展性 视图相关的可扩展性 位图形状编码 Sprite 编码 人脸动画 动态网格编码 霍夫曼编码 算术编码 容错编码	
H. 261	ITU-T	p×64kbit/s（p：1-31）	DCT 自适应量化 Z 形重排序 预测运动补偿 整数像素精度的运动估计 霍夫曼编码 容错编码	ISDN 视频会议
H. 262	ITU-T	比特率从 1.5Mbit/s 直到约 35Mbit/s	DCT 视觉量化 自适应量化 Z 形重排序 预测运动补偿 双向运动补偿 基于帧/场的运动补偿 半像素精度运动估计 空间可扩展性 质量可扩展性 霍夫曼编码 算术编码 容错编码	数字电视 数字高清电视 高质量视频 卫星电视 有线电视 地面广播 视频编辑 视频存储 数码相机

（续）

标准	标准化团体	主要目标比特率	主要压缩技术	主要目标应用
H.263、H.263+和H.263++	ITU-T	比特率从8kbit/s直到约1.5Mbit/s	DCT 自适应量化 Z形重排序 预测运动补偿 双向运动补偿 半像素精度运动估计 高级运动估计 重叠运动补偿 霍夫曼编码 算术编码 容错编码	POTS可视电话 桌面可视电话 移动可视电话 视频会议
H.264/MPEG-4第10部分	JVT（ISO/IEC和ITU-T）	比特率从8kbit/s直到150Mbit/s	整数DCT 自适应量化 Z形重排序 交替扫描排序 预测运动补偿 双向运动补偿 以小块进行的可变块大小运动补偿 四分之一像素运动补偿 图像边界运动矢量 多参考图像运动补偿 自适应内部方向预测 环路滤波器 算术编码（CABAC） 变长编码（CAVLC） 容错编码 跳跃模式 加权预测 Z形扫描、交替扫描 可伸缩视频编码 多视点编码	有线广播，地面广播和卫星广播 光磁设备、DVD、蓝光光盘上的互动或串行存储 会话类业务 视频点播，视频流 ISDN、DSL、以太网、局域网、无线和移动网络上的MMS 高清电视、超清电视、数码相机、掌上电脑、摄录像机、iPhone、iPad、无线/手持设备、电子书阅读器、工作室分布

（续）

标　准	标准化团体	主要目标比特率	主要压缩技术	主要目标应用
AVS 第 2 部分	AVS 工作组	比特率从 1Mbit/s 直到 20Mbit/s	交错处理：图像级的自适应帧/场编码（PAFF） 宏块级的自适应帧/场编码（MBAFF） 内部预测：亮度有 5 种模式，色度有 4 种模式 运动补偿：块大小为 16×16/16×8/8×16/8×8 运动矢量分辨率：1/4 像素，4 抽头内插滤波器 变换：16bit 8×8 整数余弦变换 量化与缩放：仅在编码器进行缩放 熵编码：二维变长编码与算术编码 环内去块滤波 运动矢量预测 自适应扫描	高清广播 高密度存储介质 视频监控 视频点播
AVS 第 7 部分	AVS 工作组	比特率从 1Mbit/s 直到约 20Mbit/s	内部预测：亮度有 9 种模式 色度有 3 种模式 运动补偿：块大小为 16×16，16×8，8×16，8×8，8×4，4×8 运动矢量分辨率：1/4 像素 变换：16bit 的 4×4 整数余弦变换 量化与缩放：仅在编码器进行缩放 熵编码：基于上下文的自适应二维变长编码 环内去块滤波器	移动设备上进行记录和局部回放 多媒体短信业务 流媒体和广播 实时的视频通话

（续）

标　　准	标准化团体	主要目标比特率	主要压缩技术	主要目标应用
Dirac	英国广播公司研发 Mozilla 公共许可证（MPL）	比特率从几百 kbit/s 直到约 15Mbit/s	4×4 小波变换 死区量化与缩放 熵编码：算术编码 分级运动估计 帧内帧间预测 单个及多参考帧的 P，B 帧 1/8 像素精度的运动矢量 基于 4×4 正交块运动补偿 Daubenchies 小波滤波器	广播 实况流视频 播客 对等传输 具备标清同步广播能力的高清电视 桌面产品 新闻链接 档案存储 个人视频记录仪 多级夹层编码
DiracPRO（SMPTEVC-2）	英国广播公司研发 SMPTE	比特率动无损高清到 < 50Mbit/s 压缩比 20:1	4×4 小波变换 死区量化与缩放 熵编码：基于上下文的自适应二进制算术编码，指数 Golomb 编码 帧内（也有前向、后向的预测模式） 帧、场编码（交错和渐进） Daubechies 小波滤波器	专业（高质量低时延）应用（不分配给终端用户） 面向存档的无损或视觉无损的压缩 用于现存设备再利用的夹层压缩 用于实况视频链接的低时延压缩
MPEG-7（多媒体内容描述接口）	ISO/IEC			各类多媒体信息如静止图像、图形、文本、视频、音频等，以及用于呈现多媒体的组合信息。帮助多媒体搜索引擎进行浏览、搜索和检索。视觉检索系统（例如，视频数据库、远程购物、医疗和遥感应用）、听觉检索系统（例如，卡拉 OK 和音乐销售以及历史语音数据库）。超越检索的应用（例如，代理驱动的媒体选择和过滤，以及智能多媒体呈现），还有其他的应用如教育或监控。跨模态检索系统。

（续）

标　准	标准化团体	主要目标比特率	主要压缩技术	主要目标应用
MRC（混合光栅内容）	ITU-T	最高达 33.6kbit/s	多层和多分辨率表示 每层都进行压缩算法的设计和优化	互联网彩色传真、公用电话网彩色传真、数码纸高质量文档图像压缩
VC-1（WMV-9）	SMPTE 421M	10kbit/s～8Mbit/s	整数 DCT 自适应块大小变换［（8×8）、（8×4）、（4×8）和（4×4）］ 对（16×16）和（8×8）的块进行运动估计 1/2 和 1/4 像素的运动矢量分辨率 死区和均匀量化 多种变长编码 环内去块滤波、衰落补偿	互联网上的媒体传输 广播电视 高清 DVD 剧院的数字投影、移动电话 DVB-T（地面传输数字视频广播） DVB-S（卫星传输数字视频广播）
HEVC/NGVC	JVT（ISO/IEC 和 ITU-T）		除 H.264/MPEG4 第 10 部分所列出的那些技术外，还有：率失真图像判定 RDO-Q（利用 RDO 的量化） 新偏移 自适应内插滤波器 块自适应环路滤波器（BALF） 更大的块以及相应的变换（32×32 和 64×64） 多角度帧内自适应方向预测：最多有 33 个角度的方向 帧间预测（多参考图像、双向预测、加权预测） 新的运动矢量竞争变换单元块大小（4×4 和 32×32）	HEVC（高效视频编码） 与 H.264/MPEG-4 第 10 部分相同，但比特率更低压缩效率更高 超清电视 H.264/MPEG-4 第 10 部分在较低比特率下的所有应用

（续）

标　　准	标准化团体	主要目标比特率	主要压缩技术	主要目标应用
VP6	Adobe flash（On 2技术）		I，P帧 8×8整数DCT 16×16和8×8的运动估计 1/4像素的运动矢量分辨率，去块滤波器，参考帧的最大#：2 黄金帧 预测环路滤波器 变长编码：霍夫曼和二进制算术（布尔）编码	Flash视频

表1.2　不同视频压缩标准的算法元素比较

算法元素	MPEG-2视频（H.262）	MPEG-4 AVC（H.264）	SMPTE VC-1（Windows媒体视频9）	Dirac	DiracPRO	AVS China第2部分	AVS China第7部分
帧内预测	没有：宏块编码的直流系数预测器	空间的4×4 空间的16×16 I帧-PCM	频域系数	空间的4×4	空间的4×4（前向，后向）	基于8×8块的帧内预测	帧内4×4（空间的4×4） 直接帧内预测
图像编码类型	帧 场	帧 场 图像级自适应帧场编码 宏块级自适应帧场编码	帧 场 图像级自适应帧场编码 宏块级自适应帧场编码	帧	I-帧、场（交替的、渐进的）	帧	帧
运动补偿块大小	16×16，16×8，8×16	16×16，16×8，8×16，8×8，8×4，4×8，4×4，	16×16，8×8	4×4	不适用	16×16，16×8，8×16，8×8，	16×16，16×8，8×16，8×8，8×4，4×8
运动矢量精度	全像素 半像素	全像素 半像素 1/4像素	全像素 半像素 1/4像素	1/8像素	不适用	1/4像素	1/4像素

（续）

算法元素	MPEG-2 视频（H. 262）	MPEG-4 AVC（H. 264）	SMPTE VC-1（Windows 媒体视频 9）	Dirac	DiracPRO	AVS China 第 2 部分	AVS China 第 7 部分
P 帧类型	单帧	单参考帧，多参考帧	单参考帧，亮度补偿	单参考帧，多参考帧	没有 P 帧	单参考帧，多参考帧（最多 2 个参考帧）	单参考帧，多参考帧（最多 2 个参考帧）
B 帧类型	每个方向一个参考帧	每个方向一个参考帧，多参考帧，直接及空间直接加权预测	每个方向一个参考帧	每个方向一个参考帧，多参考帧	没有 B 帧	每个方向一个参考帧，多参考帧，直接及对称模式	没有 B 帧
环路滤波器	没有	去块效应	去块效应重叠变换	没有	没有	去块滤波器	去块滤波器
熵编码	变长码	指数 Golomb 编码，CAVLC，CABAC	自适应变长码	算术编码	基于上下文的自适应二进制算术编码，Golomb 编码	二维变长编码	基于上下文的自适应二维变长编码
变换	8×8 DCT	4×4 整数 DCT 8×8 整数 DCT	4×4 整数 DCT 8×8 整数 DCT 8×4 和 4×8 整数 DCT	4×4 小波变换	4×4 小波变换	8×8 整数 DCT	4×4 整数 DCT
其他	量化缩放矩阵	量化缩放矩阵	范围缩减输入流后处理控制	量化缩放矩阵	量化缩放矩阵	量化缩放矩阵	量化缩放矩阵

1.2 视频的数字表示

视频用来或记录、或显示、或同时记录显示运动对象，它由一系列在固定时间间隔拍摄的图片组成。每秒钟的帧（图片）数叫作帧率。低于 10 帧每秒（f/s）的帧率有时候用于甚低比特率（低于 64bit/s）视频通信。对于低比特率视频通信，在 10 ~20 之间的 f/s 较为常见。采样 25f/s 或 30f/s 是电视图像的标准；而 50f/s 或 60f/s 则能产生相当平滑的运动［B8］。

视频可被分为模拟和数字类型。模拟视频用模拟信号来表示，由模拟摄像机进行逐行扫描或隔行扫描来捕获。模拟视频的例子之一就是在模拟电视制式如 PAL［G12］和 NTSC［G13］［G14］中所用的信号。数字视频常常用数字摄像机来捕获，虽然它也能由模拟视频信号转换得到。自然场景投影到数字摄像机的传感器上如电荷耦合器件（CCD）阵列，这种传感器把场景的亮度或者颜色转换成数字数据。视频序列的每幅图像（图片）包括 $M \times N$ 个像素（pixel），其中 M 是行数，N 是列数。对于彩色图像，每个像素通常包括 3 个颜色分量：红（R）、绿（G）和蓝（B）（缩写为 RGB）。每个颜色分量被单独过滤并投影到 CCD 阵列上。任意颜色都可用不同比例的 R、G 和 B 组合而成。对于 R、G 和 B 所有可能的取值，矢量（R，G，B）构成了一个空间，叫作 RGB 颜色空间。每个颜色分量用 Kbit 的整数来表示。对一般应用而言，K 值取 8 就足够了。但是，对于较为精确的应用诸如医学、广播、监控和工作室编辑等，就需要更大的 K 值。

尽管 RGB 颜色空间很适合于捕获与显示彩色图像，但由矢量（Y，Cb，Cr）形成的 YCbCr 空间对于压缩更有效，其中，Y 表示像素的亮度，Cb 和 Cr 是像素分别正比于色差 B-Y 和 R-Y 的颜色分量。如果像［B8］所定义的那样，Cb =0.546（B-Y），Cr =0.713（R-Y），那么从 RGB 空间到 YCbCr 空间的映射及其逆映射可以按如下公式来实现：

$$\begin{cases} Y = 0.299R + 0.587G + 0.114B \\ Cb = -0.169R - 0.331G + 0.500B \\ Cr = 0.500R - 0.419G - 0.081B \end{cases} \tag{1.1}$$

$$\begin{cases} R = Y + 1.402Cr \\ G = Y - 0.344Cb - 0.714Cr \\ B = Y + 1.772Cb \end{cases} \tag{1.2}$$

另一方面，如果像［G6］那样，Cb = B − Y 且 Cr = R − Y，式（1.1）和式（1.2）变为

$$\begin{cases} Y = 0.299R + 0.587G + 0.114B \\ Cb = -0.299R - 0.587G + 0.886B \\ Cr = 0.701R - 0.587G - 0.114B \end{cases} \tag{1.3}$$

$$\begin{cases} R = Y + Cr \\ G = Y - 0.194Cb - 0.509Cr \\ B = Y + Cb \end{cases} \tag{1.4}$$

人类视觉系统（HVS）对亮度误差比对色度误差更为敏感。该特性可被用来对视频数据进行更多的压缩。因此，Cb 和 Cr 分量可以用低于 Y 分量的分辨率来表示。例如，Cb 和 Cr 可以下采样为 Y 的 1/4 大小。这就降低了用来表示色度分量的数据量，同时对视觉质量没有明显的影响。对于一个随意观察图像的人来说，一幅 RGB 图像和一幅色度分辨率降低了的 YCbCr 图像之间是没有明显区别的。在存储或编码之前 RGB 图像要转换为 YCbCr 图像，而 YCbCr 图像在显示之前通常有必要转换回 RGB。一个视频标准通常支持多种 Y、Cb 和 Cr 的采样模式。典型的采样模式是 4∶4∶4，4∶2∶2，和 4∶2∶0，如图 1.1 所示。

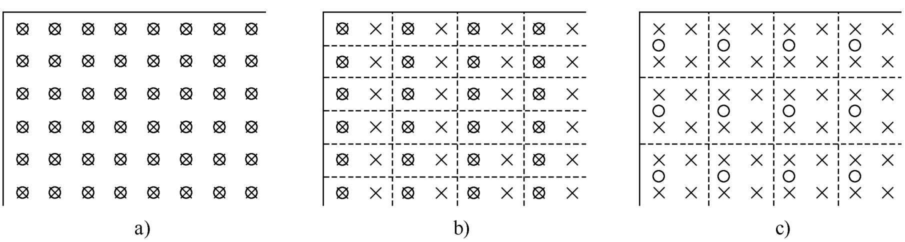


图 1.1 典型的采样模式

a）4∶4∶4 采样 b）4∶2∶2 采样 c）4∶2∶0 采样

（叉号表示亮度样本，空圈表示色度样本）

比例 N1∶N2∶N3 里面的数字指水平方向上的相对采样率，N1 表示在奇数行和偶数行里 Y 样本的个数，N2 表示奇数行里 Cb 和 Cr 样本的个数，N3 是偶数行里 Cb 和 Cr 样本的个数。例如，在 4∶2∶0 的采样模式中，N1 = 4，N2 = 2，N3 = 0。这意味着对于奇数行里的每 4 个亮度样本，有 2 个 Cb 样本和 2 个 Cr 样本；但对于偶数行里的每 4 个亮度样本，没有 Cb 和 Cr 样本。在 4∶2∶2 的采样模式中，不仅对于奇数行里的每 4 个亮度样本有 2 个 Cb 和 Cr 样本，而且对于偶数行里的每 4 个亮度样本也有 2 个 Cb 和 Cr 样本。

在以上的模式中，4∶2∶0 的采样模式是最流行的，它广泛用于消费应用领域如视频会议、数字电视和数字通用光盘（DVD）存储中。而 4∶2∶2 和 4∶4∶4 模式则用于高质量颜色再现。从图 1.1 和图 1.2 可以清楚地看出，4∶2∶0 模式的样本数仅仅是 4∶4∶4 模式的样本数的一半。

在视频编码器中，每帧图像被分为固定大小的宏块（MB），一个宏块包括一个由亮度分量 16×16 的样本和每个色度分量 8×8 的样本构成的矩形区域（4∶2∶0 格式）。图 1.2 示出 3 个视频格式 4∶4∶4、4∶2∶2 和 4∶2∶0。4∶4∶4 是全带宽视频，每个宏块包括 4 个 Y 块和 4 个 U/V 块。因全带宽之故，这种格式包括就像在 RGB

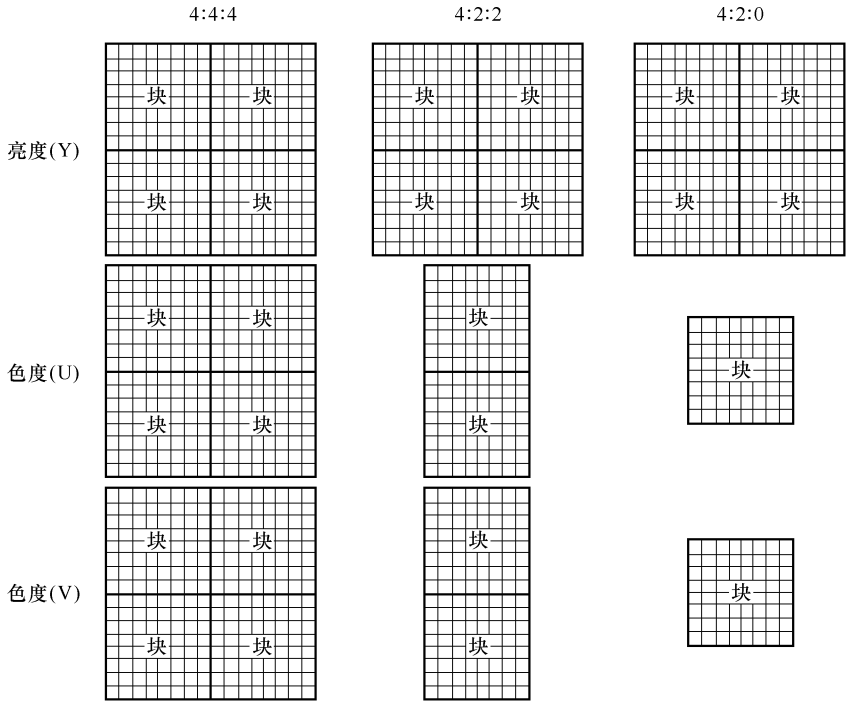


图 1.2 不同的 YUV 格式．只示出了图像的一个宏块

颜色空间中一样多的数据量。4∶2∶2 包括 4∶4∶4 一半的色度信息，4∶2∶0 包括四分之一的色度信息。4∶2∶0 格式一般用在视频流领域。

从以上讨论可以总结出：（1）数字视频由一系列图像组成，每幅图像包括 $M \times N$个像素；（2）每个像素的颜色由或者是 RGB 颜色空间或者是 YCbCr 空间中三个分量确定；（3）RGB 颜色空间更适合于图像采集和显示，而 YCbCr 空间用于编码和存储则比较好。

1.3 视频编解码的基本结构

尽管不同的视频标准之间存在着不同的差别，但是许多标准都使用了共同的工具，诸如，基于运动降低时间冗余，基于变换降低空间冗余。这些工具形成的视频编码器可以分成两类。一类包括自适应帧内预测［H32］，但另一类没有这种模式。两类基本视频编解码结构如图 1.3 和图 1.4 所示。尽管存在细小的差别，但许多视频标准包括 H.261、H.263、MPEG-1 和 MPEG-2 都具有类似的编码器结构（见图 1.3）。具有类似于图 1.4［H8］所示编解码结构的标准通常是新的视频标准，诸如 H.264、AVS China 和 VC-1［C10］［C14］。从这几幅图可以容易看出，变换工具、帧间预测、帧内预测和熵编码在视频编解码中发挥着非常重要的作用。

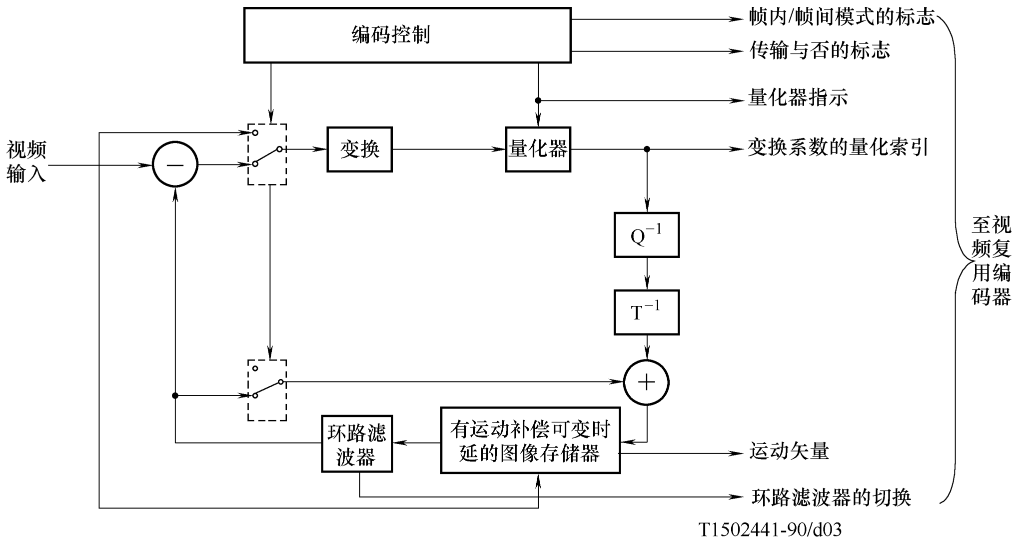


图 1.3 没有帧内预测的视频编码器

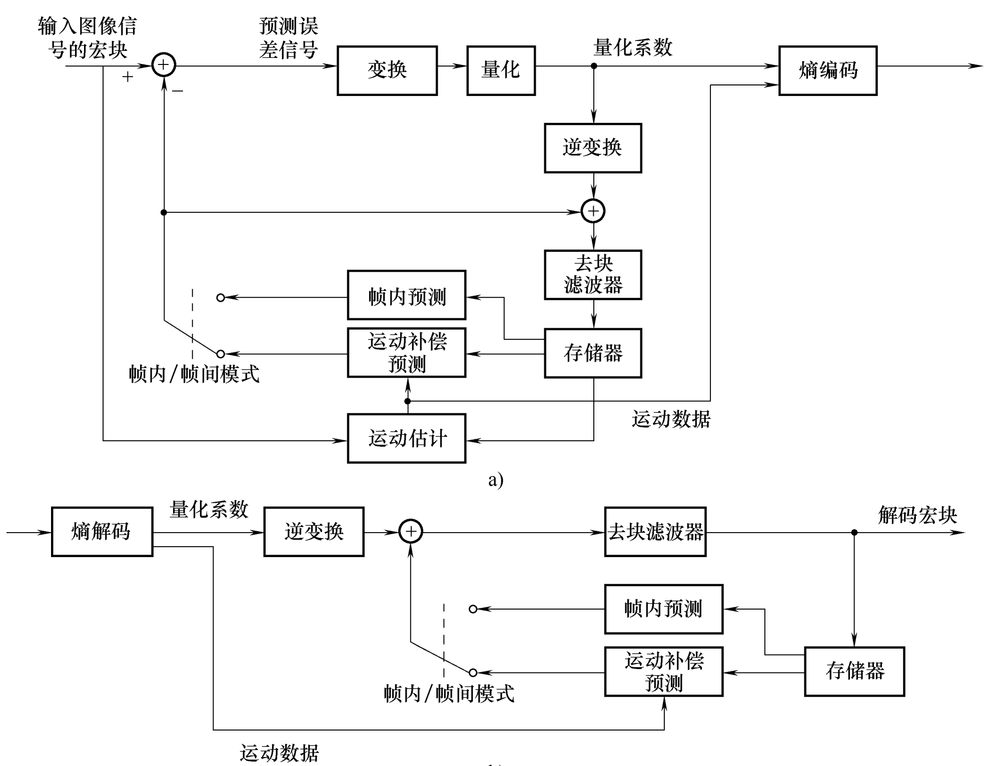


图 1.4 带有自适应帧内预测的视频编译码 [H8]

a）编码器 b）解码器

1.4 视频编译码性能比较的衡量标准

评价不同的视频编译码性能需要通过比较几个方面来进行。这些指标包括比特率（或压缩比）、计算成本（或复杂度）、质量（或失真）、可伸缩性、对错误的稳健性、和互操作性。压缩比特率是传输一个编码的视频序列需要的率。比特率的单位是每秒的比特数，简写为 bit/s（或 bps）。对一个压缩视频流计算或测量比特率是很容易的。计算成本指的是编码视频序列所需要的处理能力。

质量意味着或者主观或者客观的被测量的视频质量。平均意见得分（MOS）是衡量主观质量的标准之一。在主观试验中，一群人（通常是 15 ~ 30 个）被要求观察一组视频剪辑并对它们的质量确定等级。MOS 表示对于一个给定的视频片段对所有观察者给出等级的平均。有各种各样的主观测试方法。ITU 在各种建议［U1］-［U3］中都已正式确定了直接评级的方法。建议的测试过程包括隐式比较法诸如双激励连续质量评级法（DSCQS）、显式比较法诸如双激励损伤评级法（DSIS），或者绝对评级法诸如单激励连续质量评价法（SSCQE）或者绝对分类评级法（ACR）。有关主观测试的更多细节可以在［B11］中找到。视频质量最好是进行主观评价，也就是说由真实的观察者来进行评价。然而，主观质量的评价是很耗时而且昂贵的，这是由于需要大量的观察者和大量要被评级的视频材料；而且，对于实时系统主观评价不容易也不能例行实现。

图像或视频质量的客观评价的目的是，保持与人类质量判断一致的前提下自动评价图像或视频序列的质量。

客观度量标准可以按照不同的方式进行分类。例如，Winkler 和 Mohandas［Q21］一方面把客观度量标准分成数据度量标准、图像度量标准和基于包和比特流的度量标准，但另一方面按照所需要的参考视频信息量又分成全参考度量标准、无参考度量标准和缩减参考矩阵。数据度量标准仅仅基于逐字节的数据比较而不考虑像素的空间关系。

数据度量标准的例子是均方误差（MSE）和峰值信噪比（PSNR）。图像测度专门解释了失真和内容对于所感知质量的影响。基于包和比特流的度量标准被设计来测量网络损失对视频质量的影响，该测量基于可从无解码或很少解码的传输流和比特流中提取的参数展开。

全参考方法需要对于原始源序列和相应的处理序列进行完全的访问。对于有足够时间测量质量而且具有源视频的场合，这类方法很适合于性能测试。缩减参考方法通过从原始参考序列中提取一个参数集并用这个集合代替实际的参考视频。我们也需要以缩减参考方法传输要用的参考参数的一些途径。无参考方法只工作在处理过的视频序列上，对源信息不进行访问。缩减参考和无参考方法适合于实况监视应用［Q21］［U10］。

在过去的几十年，图像和视频质量评价受到广泛研究，制订了许多不同的主观标准［U6]-[U8］。但是峰值信噪比（PSNR）仍然是最为流行使用的质量度量标准，特别是在率失真性能的分析中。为了搞清楚 PSNR 在图像/视频质量评价中的有效范围，Huynh-Thu 和 Ghanbari［Q18］选用了十段时长 8s 的 CIF 分辨率的源（参考）视频内容（分别取名为 SRC1 到 SRC10)，这些视频内容涵盖了很宽范围的时空特性并且在 24～800kbit/s 的不同比特率上用 H.264 对其进行编码。他们还用 40 个测试序列和一个遵照国际标准的实验设置进行了主观测试［U2］。结果表明，对于特定的内容，随着比特率的增加，PSNR 总是随着主观质量单调增加，如图 1.5a 所示。换句话说，对于一个特定的编译码系统和固定的视频内容，PSNR 的变化就是质量变化的指示器。因此，在编译码的环境中，PSNR 可被用作一个性能测量标准。另一方面，图 1.5b 和 c 显示，在跨越不同视频内容评价视频质量时，PSNR 可能不是一个可靠的方法。

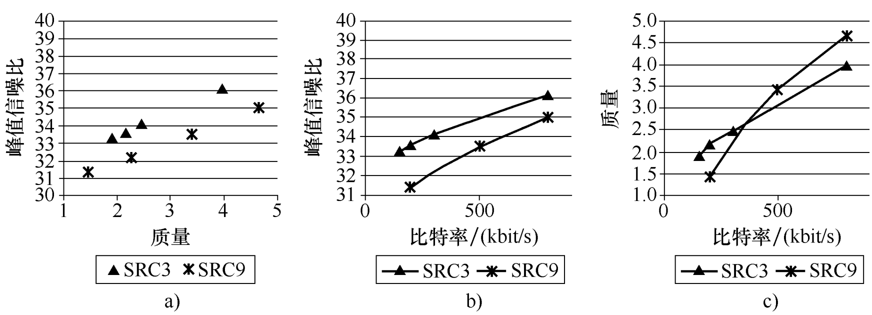


图 1.5 PSNR 和主观质量的对比

均方误差（MSE)，方均根误差（RMSE)，标准化的均方误差（NMSE)，信噪比（SNR)，和峰值信噪比（PSNR）定义如［Q4］

$$\mathrm{MSE} = \frac{\sum_{i=0}^{N-1}\sum_{j=0}^{M-1}(x_{ij}-y_{ij})^2}{NM} \tag{1.5}$$

$$\mathrm{RMS} = \sqrt{\mathrm{MSE}} \tag{1.6}$$

$$\mathrm{NMSE} = \frac{\sum_{i=0}^{N-1}\sum_{j=0}^{M-1}(x_{ij}-y_{ij})^2}{\sum_{i=0}^{N-1}\sum_{j=0}^{M-1}(x_{ij})^2} \tag{1.7}$$

$$\mathrm{SNR(dB)} = -10\log_{10}(\mathrm{NMSE}) \tag{1.8}$$

$$\begin{aligned}\mathrm{PSNR(dB)} &= 10\log_{10}\left[\frac{(x_{ij})^2_{\max}}{\mathrm{MSE}}\right]\\ &= 10\log_{10}\left[\frac{(255)^2}{\mathrm{MSE}}\right]\end{aligned} \tag{1.9}$$

式中，N 是行数；M 是列数；x_{ij}是位于第 i 行第 j 列的原始像素值；y_{ij}是位于第 i 行第 j 列处理（如解码）过的像素值。这里的 255 是 8bit PCM 的峰-峰信号值。

为了改善测量标准的测量性能，Feghali 等人［Q17］提出了一种叫作 QM 的有效的质量度量标准。QM 考虑了量化误差、帧率（FR）、和运动速度。这个度量标准 QM 定义为

$$QM = PSNR + a \times m^b \times (30 - FR) \tag{1.10}$$

式中，PSNR 是视频序列的峰值信噪比；$a = 0.986$，$b = 0.378$ 是常数；FR（≤30）表示视频的帧率；m 是运动速度参数，其实就是大的运动矢量的平均幅度归一化。产生的 QM 与评得的主观质量的相关系数（见表 1.3）在五个视频序列（Football，Ferriswheel，Mobile，Susie，and Autumnleaves）上平均高达 0.93。这要比只用 PSNR 好很多。按照式（1.10），当 FR = 30，QM = PSNR。这意味着在帧率为 30f/s 时，PSNR 很好地预测了主观质量，如图 1.6 和图 1.7 所示。

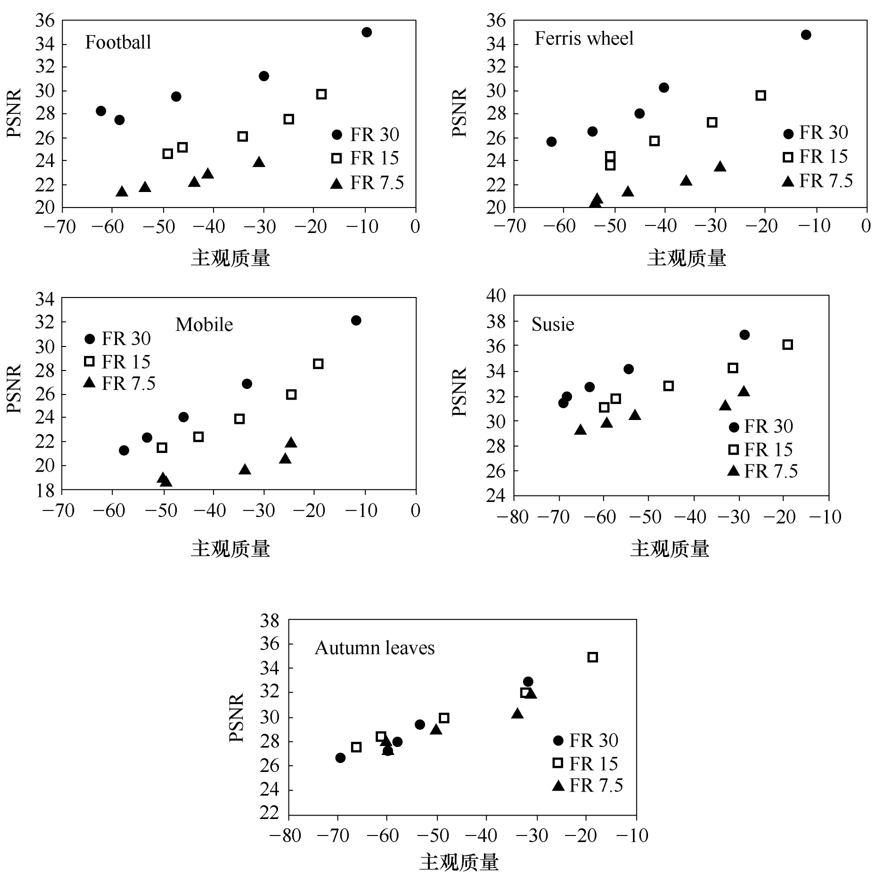


图 1.6　PSNR 与主观评价质量（解码序列和源序列的差别）［Q17］（2007 IEEE）

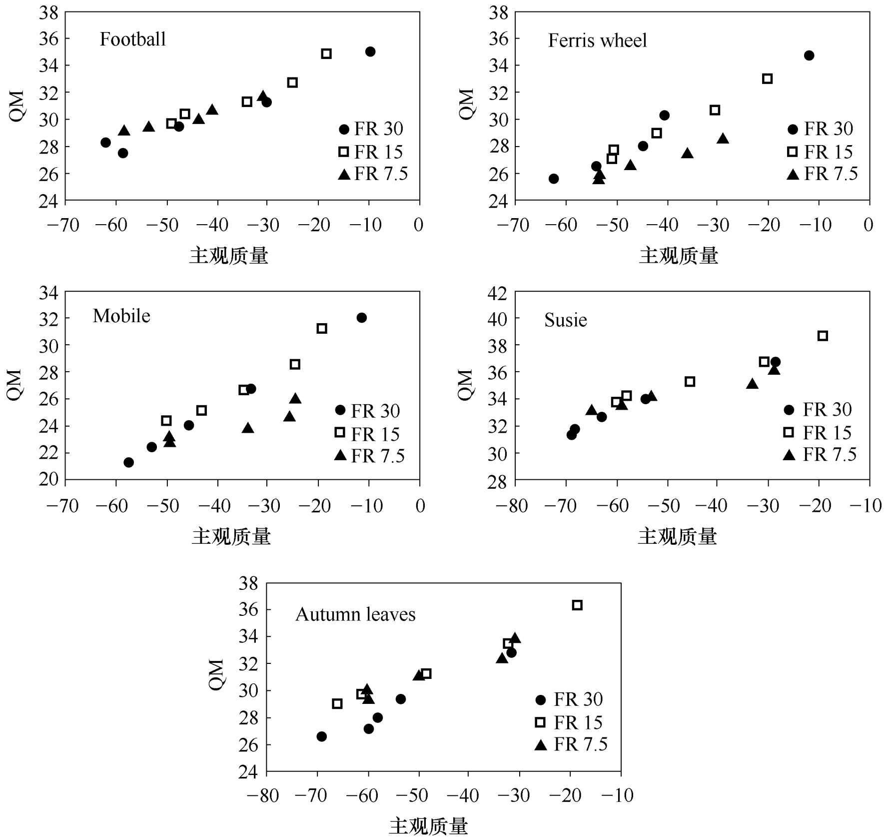


图 1.7　QM 与主观评价质量（解码序列和源序列的差别）

表 1.3　QM 与主观评价质量的相关系数［Q17］（2007 IEEE）

视频序列	QM	相关系数
Football	0.95	0.57
Ferriswheel	0.92	0.63
Mobile	0.89	0.68
Susie	0.96	0.68
Autumnleaves	0.96	0.96

与基于 PSNR 的度量标准不同，Liu 等人［Q24］提出了全参考度量标准，该标准度量由于包损失和有损压缩产生的整体质量退化。这个质量标准叫作“编码瑕疵和包损失引起的差分平均意见分的质量退化预测”（PDMOSCL，

Predicted Degradation of Differential Mean Opinion Score due to Coding-artifacts and Packetlosses)，定义为

$$\mathrm{PDMOSCL} = \mathrm{PDMOS_C} + \lambda \mathrm{PDMOS_L} \tag{1.11}$$

且

$$\mathrm{PDMOS_L} = \frac{1}{L} e^{-cL_{\mathrm{loss}}}(1 - e^{kN_{\mathrm{L}}}) \times \sum_{i=1}^{N_{\mathrm{L}}} \left(e^{-rD_i} \sum_{i=\mathrm{EL_{min}}}^{\mathrm{EL}} \alpha(\mathrm{PD}_j) \right) \tag{1.12}$$

$$\mathrm{PDMOS_C} = \frac{\mathrm{DMOS_{C,max}}}{1 + e^{s(\mathrm{PSNR} - \mathrm{PSNR}_T)}} \tag{1.13}$$

$$\alpha(\mathrm{PD}) = \begin{cases} 0, & \mathrm{PD} < \mathrm{PD_{min}} \\ \mathrm{PD} - \mathrm{PD_{min}}, & \mathrm{PD_{min}} \leqslant \mathrm{PD} \leqslant \mathrm{PD_{max}} \\ \mathrm{PD_{max}} - \mathrm{PD_{min}}, & \mathrm{PD} > \mathrm{PD_{max}} \end{cases} \tag{1.14}$$

式中，PDMOSCL 是预测的总质量退化；$\mathrm{PDMOS_C}$ 是预测的已编码序列无损部分的编码瑕疵产生的质量退化；$\mathrm{PDMOS_L}$ 是预测的包损失引起质量退化；参数 k 用来在编码瑕疵引起的质量退化和包损失引起的质量退化之间提供合适的权重；PD_j 是第 j 帧 PSNR 的下降值，$j=1$ 表示第一个丢失的帧序号；EL 是视频段损失的长度（以帧计，或叫作错误长度）；$\mathrm{EL_{min}}$是错误长度的最小值；D 是从最后一个错帧到序列尾的距离（时间，以秒计），r 是一个常数，通过主观等级到模型的最小二乘拟合来确定；L 是序列长度（帧数）；N_{L} 是包损失数量；L_{loss}叫作“损失区间”，定义为第一个损失帧到最后一个损失帧的开头（对于单个损失帧的情况，L_{loss}设为 0）；c 和 k 是待定常数；s 是 sigmoid 函数的滚降（roll-off）因数；$\mathrm{PSNR_T}$ 是 PSNR 图的过渡值；$\mathrm{DMOS_{C,max}}$是编码瑕疵引起的视觉质量退化的最大可能值。

不同场景内容的五个视频（见表 1.4）用来生成一大组测试序列。场景内容包括室内人的互动、户外运动比赛、从低到高的运动、和普通到丰富的纹理。此外，这些视频还包括各种摄像机的运动。所有的序列分辨率均为 QVGA（320×240），编码帧率是 12f/s 或 15f/s，时长为 20～40s。表 1.5 描述的四种测试可以研究质量度量标准。测试 1 和测试 2 被设计来弄清楚视觉质量是如何受到包损失影响的，而测试 3 关注编码瑕疵的影响。前 3 项测试所用的序列都从原始视频“American Pie”中产生，用来探索和训练客观度量标准 PDMOSCL。测试 4 所用的序列从全部五个视频源中生成，兼有编码和包损失瑕疵，用来验证 PDMOSCL。结果表明，对于图 1.8所示的不同损失类型、编码瑕疵、和场景内容的一大组视频片段，这个度量标准与主观质量级别具有很好的相关度。

除了以上的全参考测度，Ninassi 等［Q23］也描述了一个全参考视频质量评价测度，它专注于空间失真的时间变化。该空间失真的时间变化在整个视频序列上以眼睛注视水平进行评价。

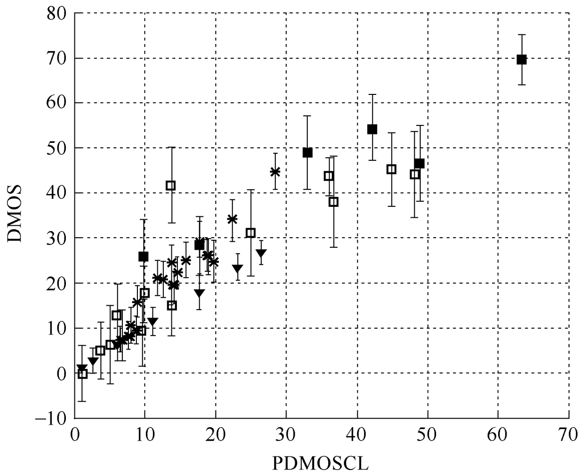


图 1.8 DMOS（差分评价意见分）和 PDMOSCL 的关系

（"星形"表示测试 1 和 2 中的数据，"三角形"表示测试 3 的数据，实心"方形"对应测试 4 中的训练序列。空心"方形"用于检验测试 4 中的序列，［Q24］2009 IEEE）

Pinson 和 Wolf［Q14］提出一个通用的视频质量测度（VQM），定义如下

$$\begin{aligned}\text{VQM} = & -0.2097 \times \text{si_loss} + 0.5969 \times \text{hv_loss} + 0.2483 \times \text{hv_gaint} \\ & + 0.0192 \times \text{chroma_spread} - 2.3416 \times \text{si_gain} + 0.0431 \times \text{ct_ati_gain} \\ & + 0.0076 \times \text{chroma_extreme} \end{aligned} \quad (1.15)$$

这里，si_loss 检测空间信息的下降或损失（例如，抖动）；hv_loss 检测从水平和垂直方向到对角方向的边缘偏移，例如可能是这种情况：水平和垂直边沿比对角边沿有更多的抖动；chroma_spread 检测二维颜色样本分布的范围变化。这样，chroma_spread 度量了颜色的损伤。si_gain 度量由边沿锐化或增强产生的质量改进；ct_ati_gain用对比度特征和时间信息特征的乘积来计算，前者度量空间细节，后者度量 S-T 区域出现的运动量；chroma_extreme 检测严重的局部颜色损伤，诸如由传输误差产生的损伤。

表 1.4 测试视频的内容描述［Q24］（2009 IEEE）

视频名称	内容描述
Americanpie	电影预告片，人们在谈话和走动
F1carrace	奔驰的汽车，繁忙的车站人群
Interview	人们在静静地坐着，谈话，近看
Paris	人们在静静地坐着谈话，有人体的运动
Baseketballgame	运动员在奔跑，射球和庆祝

表 1.5 四项测试简述［Q24］（2009 IEEE）

测试 1（12 个序列）		测试 2（13 个序列）	
序列描述	核查因数	序列描述	核查因数
在不同位置有相同误差的 3 个序列	损失位置	位置不同误差相同的 3 个序列	损失位置
不同误差长度的 3 个序列（GPO = 2s）	误差长度	短误差长度的 2 个序列	误差长度损失能见度
不同损失类型的 3 个序列	损失类型	PSNR 下降不同的 5 个序列	损失的严重程度
不同损失量的 3 个序列	损失量	不同误差类型的 3 个序列	损失类型
测试 3（7 个序列）		测试 4（25 个序列）	
序列描述	核查因数	序列描述	核查因数
以不同 QP 编码的 7 个序列	PSNR	以不同 QP 编码，包损失随机的 25 个序列	模型验证

VQEG FR-TV（视频质量专家组全参考电视）第二阶段测试［U9］已经证明，该模型可以对 525 行和 625 行的视频系统产生极好的视频质量估计，如图 1.9 和图 1.10所示。

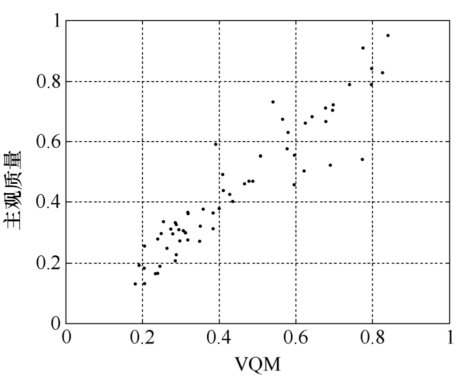


图 1.9 525 行 VQEGFR-TV 第二阶段测试数据：片段主观质量与片段 VQM［Q14］(2004IEEE)

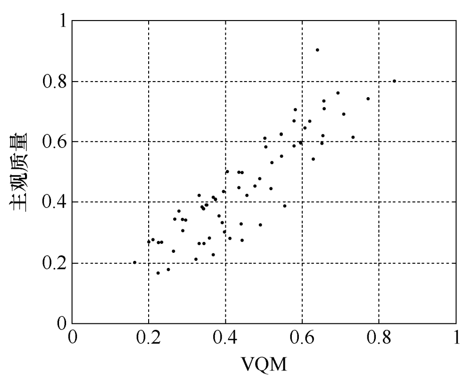


图 1.10 625 行 VQEGFR-TV 第二阶段测试数据：视频段主观质量与视频段 VQM［Q14］(2004IEEE)

无参考视频质量评价已经成为视频质量测度领域中许多研究的主题。Oelbaum 等［Q25］提出了一种无参考测度，用到 7 个特征包括抖动、块效应、空间活动性、时间可预测性、边沿连续性、运动连续性、和颜色连续性。如果这些特征值分别用 p_i（$i=1，2，\cdots，7$）来表示，再定义特征矢量 $p=(p_1，p_2，\cdots，p_7)^{\mathrm{T}}$，那么 NR 质量测度 $\hat{y}$ 可被写作

$$\hat{y}=b_0+p^{\mathrm{T}}b \tag{1.16}$$

式中，b 是包含每个特征 p_i 的单独估计权重 b_i（$i=1, 2, \cdots, 7$）的列矢量；标量 b_0 是模型偏移。四个不同的特征类型叫作低比特率、抖动、块效应和常规类型，被定义来对视频序列进行分类。对每个视频类型这 4 个不同模型只是特征 p_i 的权重 b_i（见图 1.6）不同。

表 1.6　不同特征类型的权重［Q25］（2009 IEEE）

	低比特率	抖　动	块效应	常　规
抖动	-0.019	-0.040	-0.060	-0.030
块效应	-0.046	-0.090	-0.114	-0.045
空间活动性	0.024	0.078	0.060	0.067
可预测性	0.003	0.015	0.026	0.020
边沿连续性	0.015	0.046	0.040	0.024
颜色连续性	0.006	0.029	0.014	0.019
运动连续性	0.034	0.031	0.022	0.074

对于给定的视频序列 V，通过分析视频的低质量版本 V_{low} 的特征选择一个合适的模型，V_{low} 是通过用一个高的量化参数（QP）对实际视频 V 进行编码产生的。然后，估计 V 的特征值，选出的测度模型用于计算无参考质量测度。欲知更多细节，请参考论文［Q25］。

Naccari 等［Q26］提出了一个无参考视频质量测定算法（称为 NORM）来自动评价视频序列中信道引入的失真，该序列从 H.264/AVC 兼容比特流中解码得到，而该比特流是经由一个受到包损失影响的有噪信道进行传输的。但是 NORM 只是度量信道损失带来的失真。

与主观质量具有很好相关度的一个客观测度是结构相似度指标测度 SSIM［Q13］，在附录 C 中有详细描述。

1.5　音频的数字表示

数字化表示音频信号最普通的方法是用脉冲编码调制（PCM）原理将一个模拟音频信号数字化［O1］。以固定间隔对每个通道的音频信号进行采样，然后用 A/D 转换器将其数字化为 PCM 码，该码实际上是离散数字［O7］［O4］。按照奈奎斯特采样定理［B20］，如果对一个信号在固定间隔以略高于信号最高频率两倍的速率进行即时采样，那么样本值就包含原始信号的所有信息。音频数据采集的采样频率应该按照人类听觉系统的特性来确定。人耳的动态范围大约是 140dB，听力带宽最高为 20kHz［B15］［O32］。因此，音频信号应该以至少 40kHz 的采样率进行采样才能获得高质量音频。这就是 CD 格式采样率为 44.1kHz（或者 44100 个样本/s）的原因，这个速率比我们能听到的最高频率的两倍稍好一些。为了避免混叠

噪声，模拟音频信号必须通过采样-保持［O3］电路和 A/D 转换器前置的低通滤波器限制带宽［O3］。另一个重要的因素是 A/D 转换器的分辨率（比特数），它确定了音频系统的动态范围。16bit 的分辨率只够再生动态范围为 96dB 的声音，而 24bit 理论上可实现 144dB 的动态范围。典型的采样率和分辨率分别包括 44.1 –，48 –，96 –，192-kHz，和 16 –，20 –，24-比特。DVD 单声道和立体声支持所有这些采样率和分辨率［O33］，而 CD 音频在 16bit 上支持 44.1kHz。16 比特分辨率上的 48kHz 采样率能产生 768kbit/s 每信道的数据率，对于立体声信号这意味着大约 1.5Mbit/s，如图 1.11 所示。

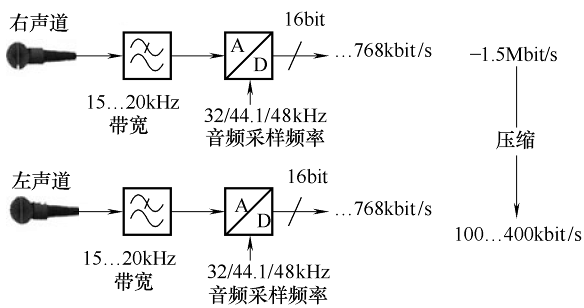


图 1.11 数字音频表示［B15］（2008Springer）

1.6 感知音频编码的基本结构

音频编码的核心目的是用最小的比特数来表示信号，同时实现透明的信号再生，也就是说，即使对于一个耳朵灵敏的听者，产生的输出音频与原始输入也无法区别［O18］。感知音频编码在音频编码中发挥着重要的作用。人们提出了许多感知音频编码方法［O13，O18，O20，O26，O28-O30］和若干标准［O8-O12，O17，O19，A4］。多数感知音频编码器都相当类似于图 1.12 所示的结构。

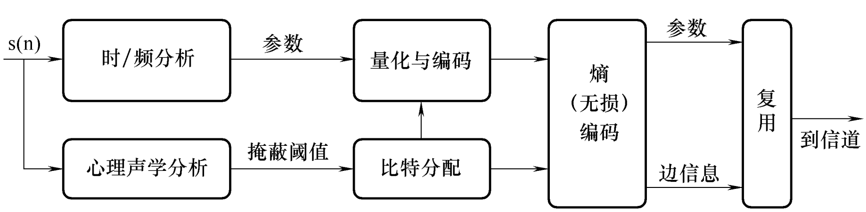


图 1.12 一般的感知音频编码器［O18］（1997IEEE）

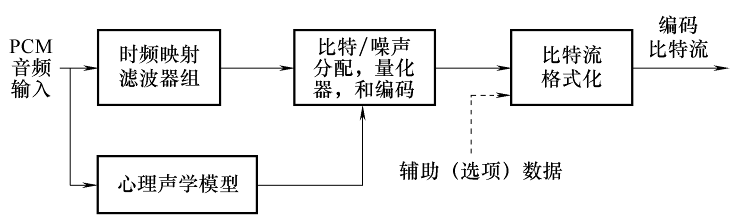
编码器一般把输入信号分割为时长为 2～50ms 的准平稳帧。然后时/频分析部分对每个分析帧进行分解。时/频分析近似于人类听觉系统的时间和谱分析特性。它把输入音频变换为一组参数，这组参数可以按照感知失真测度进行量化和编码。

根据系统的总体目标和设计思想，时/频分析部分可以包含如下之一：

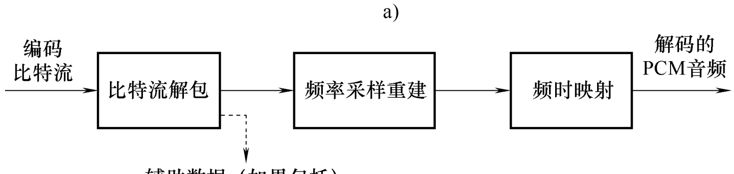
- 酉变换；
- 时不变均匀带通滤波器组；
- 时变（信号自适应的）临界采样非均匀带通滤波器组；
- 混合变换/滤波器组信号分析器；
- 谐波/正弦分析器。

时-频分析方法的选择总是涉及时间和频率分辨率需求之间的基本权衡。感知失真控制由心理声学信号分析部分来控制，该部分基于心理声学原理估计信号的掩蔽功率。心理声学模型传送确定最大失真量的掩蔽阈值；在量化和时频参数编码过程中，该失真可加入到时间-频率平面上的每一点上，而没有在重建信号中引入可听到的瑕疵。心理声学模型因而允许量化和编码部分在时频参数集中利用感知不相关性。量化和编码部分也能通过经典方法诸如差分脉冲编码调制（DPCM）或自适应DPCM（ADPCM）利用统计冗余。量化可以是均匀的或者是概率密度函数优化的（Lloyd-Max），而且可以对标量或矢量（VQ）进行。一旦形成了一个紧凑的量化参数集，剩下的冗余一般通过游程和熵编码（例如，霍夫曼，算术，LZW（Lempel-Ziv-Welch）[DC2]）技术来去除。因为心理声学失真控制模型是信号自适应的，故大多数算法都具有内在的可变率。通常通过缓存反馈方案来满足固定信道速率的要求，这样做常常会引入编码延时。

MPEG音频编译码和AVS音频编译码分别如图1.13和1.14所示。我们可在这些标准的编译码中很容易发现类似于图1.12中的基本结构。有关MPEG和AVS音频的更多细节可以分别在［O11，O12，O19］和［N2，N5，N8］中找到。



a)



b)

图1.13 MPEG音频编译码［O12］（1995 IEEE）

a）编码器 b）译码器

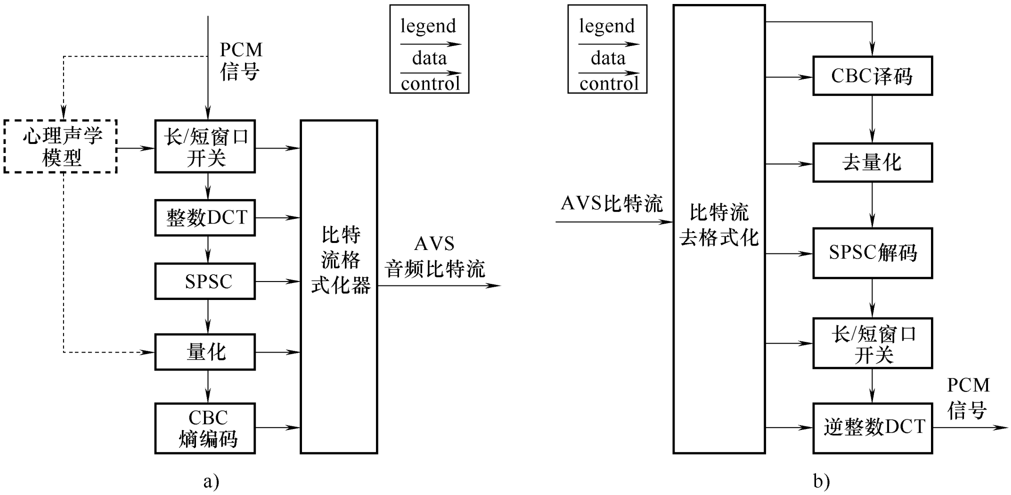


图 1.14　AVS 音频编译码［N2］（IEEE 2005）

a）编码器　b）解码器

1.7　音频编译码的性能比较测度

像视频编译码的性能评价一样，为了评价音频编译码的性能，也应该考虑一些方面包括每个声道的比特率、计算成本（或复杂度）以及质量。与视频一样，比特率的单位是每秒的比特数（bit/s）。

音频质量评价方法可以分为两类：主观方法［L3-L6，L8，L16］和客观方法［L7，L10，L11，L15，L17，L19，O27］。ITU-R 推荐方案 BS.1116［L4］定义了一个高质量音频编译码主观评价的测试程序。该测试程序基于“双盲、隐含参考三重激励”对比方法。在此方法中，给定听者三个激励信号：参考（原始）信号、测试信号 A 和 B。在 A 和 B 当中，有一个损伤（编码）信号，另一个也是原始信号（隐含参考）。但是测试对象和监督者都不知道 A 和 B 当中哪个信号是隐含参考。在听完所有的三个信号，该对象必须从 A 和 B 中挑出隐含参考，然后利用如图 1.15 所示的五级损伤标准相对于参考激励对另一个（编码信号）进行分级。从最好到最坏，编码失真分级为“不可察觉的（5）”，“可察觉但不讨厌的（4.0 ~ 4.9）”，“轻度讨厌的（3.0 ~ 3.9）”，“讨厌的（2.0 ~ 2.9）”，或“非常讨厌的（1.0 ~ 1.9）”，见图 1.15。

如果用符号 SDG、G_i 和 G_r 分别来表示主观差分等级、实际损伤信号的得分和实际隐含参考信号的得分，那么 SDG 可定义为

$$SDG = G_i - G_r \tag{1.17}$$

G_r 的默认值是 5。当主体正确地识别了隐含参考，SDG 为负值；如果主体识别

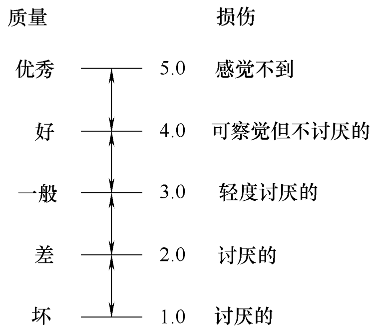


图 1.15　五级损伤标准

错了隐含参考，SDG 为正值。经过多个对象和多次试验，计算出平均损伤分数，再相对于理想情况计算编译码性能。

在 ITU-R 的推荐方案 BS.1534［L8］中，开发了一个叫作 MUSHRA［L16］(具有隐含参考和锚的多激励）的主观方法用于编码系统中等质量水平的评价。MUSHRA 是一个双盲多激励测试方法，有一个已知参考、一个隐含参考和一个或多个隐含锚。至少需要一个锚是参考信号的低通版本。按照 MUSHRA 原则，需要参试对象以连续质量等级制对激励进行评级，该等级分为五个区间，从上到下标记为优秀、好、一般、差和坏。然后这些分数正规化范围为 0～100，这里 0 对应最低等级（坏质量)。

因为主观测定既耗时又昂贵，人们开发了多种客观音频质量评价方法用于音频质量的自动评价。音频质量评价的综述和最新进展如［L19］陈述，其中包括音频质量感知评价（PEAQ）算法的简要技术总结，该算法在 ITU 标准 BS.1387［L7］中提出。

设计 PEAQ 只是为了客观地对有极小损伤的信号进行分级。图 1.16 所示的框图描述了 PEAQ 的两个部分：心理声学模型和认知模型。心理声学模型包含许多不同的模块，它们对人类听觉系统的各个独立部分建模。它把时域输入信号变换为一种颅底膜表示（即，人类听觉系统中颅底膜的模型)。认知模型模拟了人脑的认知过程。它处理由心理声学模型产生的参数来形成一个质量分数。图 1.16 也告诉我们 PEAQ 是一种侵入算法，它通过比较两个输入信号：一个参考（原始）信号和一个退化（编码）信号来产生一个分数作为质量测度。

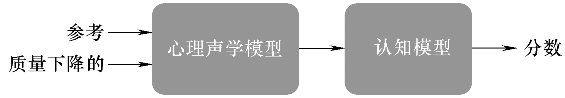


图 1.16　PEAQ 算法中两个主要部分的框图［L19］（2009 Elsevier）

PEAQ 包括两个版本：一个基本版本和一个高级版本。前者用在计算效率很成问题的场合，后者用在准确度至关重要的场合。基本版本和高级版本的主要结构差异在于，基本版本只有一个外耳模型（基于 FFT［B12］的耳朵模型）而高级版本有两个外耳模型（基于 FFT［B12］的和基于滤波器组的耳朵模型）。基本版本产生 11 个模型输出变量（MOV，Model Output Variables）而高级版本只产生 5 个 MOV。MOV 是基于响度、调制、掩蔽和自适应性的输出特征。MOV 输入到神经网络并训练网络把他们映射到一个单独的 ODG（overall difference grade，总体差分等级）分数。ODG 分数预测了退化信号的感知质量。ODG 分数范围为 0 ~ −4，0 表示信号具有不可感知的失真，−4 表示信号具有非常讨厌的失真。

基于 FFT 的耳朵模型用在 PEAQ 的两个模型中，在频域处理样本帧。基于滤波器组的耳朵模型只用于 PEAQ 的高级版本中，在时域中处理数据。在图 1.17 中所示的认知模型中，使用神经网络把 MOV 映射到一个单一的 ODG 质量分数。而产生 MOV 需要用到两个耳朵模型的输出。有关基于 FFT 的耳朵模型和基于滤波器组的耳朵模型的更多细节，请参见［L7］和［L9］。

参数诸如响度、调幅、适应性和掩蔽参数的范围决定了 MOV。这些 MOV 也对一些概念如线性失真、带宽、NMR、调制差异和噪声响度进行了建模。它们通常以这些参数的平均值来计算，这些参数取自于测试和参考信号的持续时间段内；一般而言，要从每类参数（调制、响度、带宽等）中得到不止一个 MOV。有关 MOV 的详细描述在［L7］和［L19］中可查。

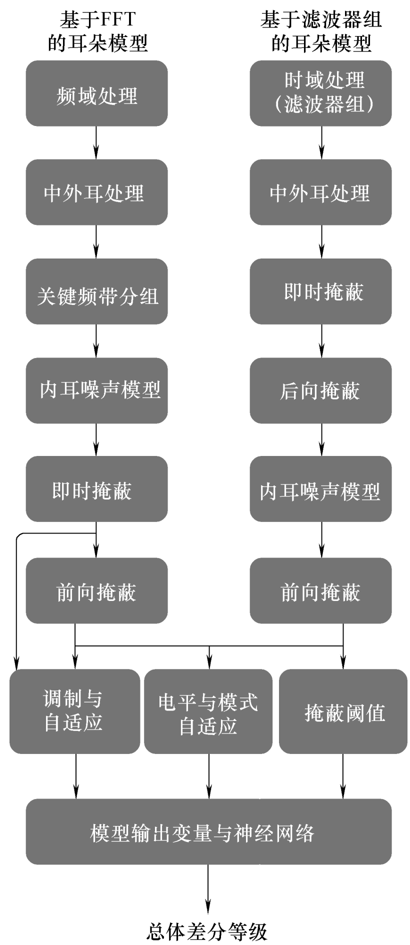


图 1.17　PEAQ 的详细框图［L19］（1995 Elsevier）

式（1.18）定义的分段 SNR［L1，L2，L12，L18］是一种简单的客观语音质量亮度，它是多个短片段 SNR（信噪比）值的平均。

$$\mathrm{SNR}_{\mathrm{seg}} = \frac{10}{M_{\mathrm{B}}}\sum_{m=0}^{M_{\mathrm{B}}-1}\log_{10}\left(\frac{\sum_{n=N_m}^{N_m+N_{\mathrm{s}-1}} x^2(n)}{\sum_{n=N_m}^{N_m+N_{\mathrm{s}-1}}[d(n)-x(n)]^2}\right) \tag{1.18}$$

式中，$x(n)$ 表示原始语音信号；$d(n)$ 表示失真信号；n 是样本下标；N_s 是片段长度；M_B 是语音信号中的片段个数。Mermelstein［L1｝以另一种方式定义了分段 SNR。以 SNR_{seg}进行的性能测量是一个波形编译码声音质量的很好的估计量，但对于语音编码器它的性能比较差，因为这里的目的是生成同样的语音而不是产生语音波形本身。而且，分段 SNR 与主管知觉质量的相关度很低（只有 0.531，如［L17］中给出）。尽管分段 SNR 不适合独立作为知觉音频编码器的性能评价测度，但通过与其他分量进行线性组合构造出新的性能优越的知觉质量测度时它可用作一个分量［L17］。

Kandadai 等人［L14］将评价结构相似度应用于音频知觉质量的客观评价。有两种方法。在第一种方法中，音频序列分成长度为 128 重复率为 50% 的时间帧，然后分别对每一帧使用结构相似度（SSIM）［Q13］。通过对每帧的 SSIM 取平均算出平均 SSIM（MSSIM）。这种方法被称作时间 MSSIM（T-MSSIM）。在第二种方法中，使用了 50% 的重复窗口 256 点改进的离散余弦变换（MDCT）［O5，O6］将音频序列分解为一种时频表示，然后把 SSIM 用于时频表示的二维块。这种方法被称作时频 MSSIM（TF-MSSIM）。T-MSSIM 和 TF-MSSIM 与 MUSHRA［L8］主观质量的相关系数分别是 0.98 和 0.976。这说明 MSSIM 与主观测试高度相关。

［L9］提出了能量均衡质量测度（EEQM）用于高损伤音频的质量评价。在该方法中，原始音频谱图以阈值 T_{EEQM}进行截取（原始音频谱图系数超过 T_{EEQM}的幅值保留，其他的设为零）。对于 TEEQM 的每个特定的值，计算截断谱图的能量并与重建信号的带通谱图能量进行对比。TEEQM 通过一个迭代优化算法进行调整，使原始频谱的截断版本与重建信号的带通谱图具有相等的能量和类似的时频特性。然后，最优的 TEEQM 用作测试信号的损伤度量。而且，TEEQM 与模型输出变量（MOV）［L7］组合在一起形成一个简单又稳健的音频质量通用测度［L13］。

谱带复制（SBR）［O22，O23］是一种新的音频编码工具，它显著改善了知觉编码器和语音编码器的编码增益。目前，通过结合 SBR 有三个不同的音频编码器表现出极大的性能改善：MPEG-AAC，MPEG 第 2 层和 MPEG 第 3 层（mp3），这三个都是开放标准 ISO-MPEG 其中的部分。AAC 和 SBR 的组合将用在标准化数字无线电 Mondiale（DRM）系统中，SBR 目前正被标准化在 MPEG-4 内。SBR 就是所谓的带宽扩展技术，其中信号带宽的主要部分由接受边带的下边带重建得到。论文［O23］重点关注了 SBR 的技术细节，特别是滤波器组，它是 SBR 过程的基础。

AAC 和 SBR 的组合——aac+，是目前最有效的音频编码器，与已经很强大的 AAC 编码器相比，编码效率提高了 30% 以上。SBR 系统的基础是调制的复数 QMF 组，复数表示允许子带修改而不会引入过多的混叠。

1.8 总结

本章简述（对比）了不同的视频/图像编码标准，诸如 JPEG，MPEG 和 H. 26x 系列之外，还有 DIRAC（第 7 章）、AVS China（第 3 章）和 VC-1（第 8 章）。对比测度紧接着关于音频编码的类似简述总结了本章。这些方面在第 2 章做进一步的阐述。

第 2 章　视频编码标准和视频格式

摘要：介绍了视频格式以及 RGB、Y、Cb、Cr 和 YUV 之间的转换。这些基本上是第 1 章的延续，因而补充了第 1 章讨论过的内容。

关键字：视频压缩　视频编码标准　采样格式　视频格式　RGB　YUV　YCbCr　质量　PSNR　SSIM

2.1　前言

从模拟电视到数字电视，从 VHS 视频磁带到 DVD，从只用于打电话和发短信的蜂窝电话到具有照相、网络浏览器、导航系统、社交网络平台功能而很少用于打电话的蜂窝电话，在过去几年中按照用户创造、分享和观看视频的方式，已经经历了一场重大的变革。商业因素和技术进步推动了数字视频工业的不断演进。商业驱动来自于用户和商家对于潜在巨额收益的信心。在技术领域，这些因素包括较好的通信基础设施、廉价的宽带网络、4G 移动网络以及便于使用的记录、编辑、共享和观看视频的应用程序。

在从信源（摄像机或存储的视频片段）获得视频到目的终端（显示器），涉及一系列过程。这项工作的关键过程是压缩（编码）和解压（译码），然后重建来显示，编译码涉及把“带宽密集型的”原始视频源降低到最优大小以适于传输或存储。为了使商业和技术优势进入到产品中，压缩和解压过程应该在三个参数之间达到一种合理的平衡，这三个参数彼此是不相关的：视频质量、编码时间和大小。因此人们对于视频压缩和解压技术和系统方面存有浓厚兴趣。

高比特率连接、大容量硬盘、闪存和光介质的市场经历长期发展持续壮大已经充分满足用户的需求。伴随着传输或存储每个比特的价格的不断下降，视频压缩成为绝对必不可少的工作，而且人们进行了大量的努力使其更加完善。让我们想象一下没有视频压缩的世界，当前的互联网吞吐率将不够实时地处理未压缩的视频（甚至在低帧率和/或小的帧分辨率下），数字通用光盘（DVD）只能以电视质量的分辨率和帧率存储几秒钟的原始视频。视频压缩使得资源的有效传输和存储成为可能。例如，如果具备高比特率传输信道，那么发送高分辨率压缩的视频或多个压缩视频比发送单个低分辨率和无压缩的视频流更有吸引力。即使伴随着存储和传输能力的稳步发展，在未来很长的时间里，压缩仍然有可能是多媒体业务中必不可少的一个部分。

按照定义，压缩是从一个携带信息的信号中去除冗余的过程。在无损压缩系统中，去除统计冗余从而使原始信号能在接收端被完美地重建。不幸的是，这里存在一

种折衷：无损方法只能达到视频信号不怎么高的压缩量。大多数实用的视频压缩算法都是基于有损压缩的，可以实现更大的压缩，但后果是解码的视频信号与原始信号不完全相同。视频压缩算法的目的是实现有效压缩的同时，使压缩过程引入的失真最小。

对于视频片段的情况，有可能通过将无损和有损编码背后的原理组合起来进行数据的压缩。构造视频片段最简单的办法是截取连续几幅图像并把它们称为帧。在视频片段里存在许多固有的冗余，一个给定的帧里包含的多数信息也会出现在前一帧中。任何帧内只有一小部分是新信息；通过计算那一小部分信息的所在，而且只存储那部分信息量，那么就有可能大幅降低该帧的数据量。压缩过程涉及将压缩算法用于源视频以创建一个压缩文件以便于传输或存储。将相反的算法用于压缩视频以产生一个看起来具有与原始视频几乎相同的内容的视频。这对算法一起工作，叫作视频编解码器（Codec，编码器/解码器）。

视频压缩算法如 MPEG-4［B8］和 H.264［B8，B18，H44］是高度复杂的过程，涉及许多技术，如差分编码，其中只有第一帧是被完全编码的。如图 2.1 所示，在两幅连续的图像中，作为参考的第一幅图像充当静态元素，也就是房子。只有运动部分，也就是奔跑的人，用运动矢量进行编码，这样就降低了发送和存储的信息量。也包括一些技术如基于块的运动补偿来进一步减少数据。基于块的运动补偿基于如下观察：视频序列中新的一帧可以在前面一帧中找到，不过可能在不同的位置。该技术把一帧分为一系列宏块（像素块）。新的一帧可以通过逐块地在参考帧中发现匹配块的方式预测得到。如果存在匹配，编码器只对参考帧里发现匹配块的位置进行编码。这个方法比编码一个块本身的内容需要少得多的比特（见图 2.2）。

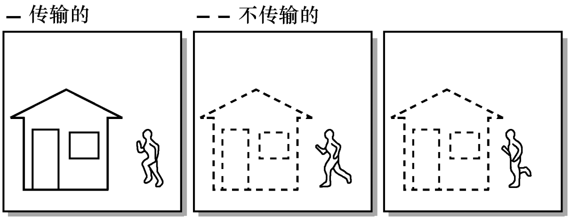


图 2.1　运动视频压缩算法的帧间预测［V9］

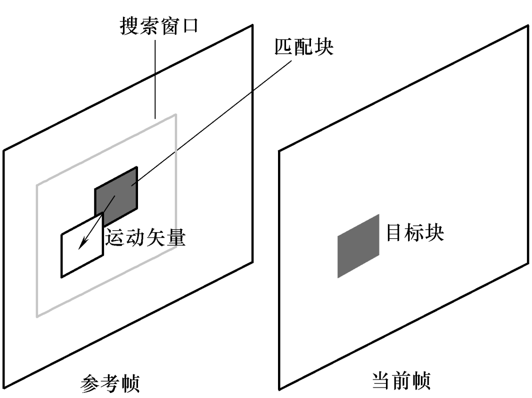


图 2.2　基于块的运动补偿图解［V9］

H. 264 视频编码标准是由 ITU-T 视频编码专家组（VCEG，Video Coding Expert Group）和 ISO/IEC 运动图像专家组（MPEG，Moving Picture Expert Group）[H44] 联合开发的最新的基于块运动补偿的编译码标准。H. 264 标准的目的是建立一个在低得多的比特率上能实现与以前标准质量相当的标准。H. 264 提供了比以往任何标准更好的压缩，它包括若干内置的特征以支持在一系列信道和网络上进行可靠稳健的传输。不幸的是，与以往标准相比，这需要付出增加编码器计算复杂度的代价。为了实现 H. 264/AVC 的实用性，必须在保持编码效率的同时实现编码复杂度的显著降低 [H35]。

2. 2　复杂度的降低

在智能手机和平板电脑等设备上捕获和播放高清视频的需求导致了开发有效的低复杂度视频编码器这一颇具挑战性的工作。来自全球的研究者们提出了许多方法来降低 H. 264 的复杂度。人们提出了多种不同的帧内模式复杂度降低方法，如 [H10，H12，H14，H35]，但是很少有方法实现了有效的编码。一些方法减少了编码时间，但是不能保持原始视频片段的编码质量。在编码增益和视频质量之间达到一种平衡是很重要的。[H27] 提出了一种有效的帧内编码模式复杂度降低算法；大大降低了视频片段的编码时间，带来的质量损失和比特率的增加却都可忽略。

Muniyappa 的学位论文（见最后的附言）通过利用 JM 18. 0 [H30] 集中讨论了降低 H. 264 帧内模式选择的编码复杂度。它很大程度上基于相邻宏块特性趋于类似这一现象。这样，通过简单地使用方向掩模和相邻模式，检查所有可能编码模式组合的 RDO（率失真优化）过程通常繁重的任务能得以显著减轻 [H27]。按照不同视频格式和视频上下文的编码时间来看，实验结果表明了复杂度的减少。2. 3 ~ 2. 5节给出了有关视频编码标准和视频格式的简要但深入的陈述。

2. 3　视频编码标准

人们已经提出了许多视频编码方法，而且许多其他研究仍在进行中。每年发表的数以百计的研究论文描述了各种新的压缩方法。然而，商业视频编码应用趋向于使用有限的几个标准化方法进行视频压缩。标准化视频编码格式具有如下若干好处 [B14]：

- 标准简化了来自不同出品商的编码器和解码器之间的互操作性；
- 标准使建立综合视频平台成为可能，在次平台中，许多不同的应用如视频编译码、音频编译码、传输协议、安全和版权管理以定义完好且一致的方式进行互动；
- 许多视频编码方法都获得了专利权，因此一个特定视频编译码执行方案可

能会存在侵犯专利权的风险。执行标准所需要的技术和算法都是严格定义的，而且涵盖这些技术的专利许可费，即授权使用包含在专利中的技术的费用，也被清楚地规定下来。

2.4 MPEG 和 H.26x

这些建议或国际标准是由 ITU-T SG16 Q.6（国际电信联盟）[也叫作 VCEG（视频编码专家组）] 和 ISO/IEC JTCI/SC29/WG11（国际标准化组织）[也叫作 MPEG（运动图像专家组）] 联合准备的。VCEG 在 1997 年 [H44] 形成，以维护以前的 ITU-T 视频编码标准并开发新的用于一系列会话和非会话业务的视频编码标准。MPEG 于 1988 年 [S13] 形成，以建立面向各种应用如数字存储媒体、发布和通信的运动图像和相关音频的编码标准。以后，在 2001 年，ITU-T 视频编码专家组（VCEG）和 ISO/IEC 运动图像专家组（MPEG）形成了一个联合视频团队（JVT）以开发一个新的建议或国际标准，H.264 建议/MPEG-4 标准第 10 部分 [H12]。

2.4.1 H.120

H.120 [S1]，是在 1984 年由 ITU-T 以前的 CCITT（国际电报电话咨询委员会）开发的第一个数字视频编码标准。之后它演化出不同的版本，1984 年开发的版本 1 的特征是条件补充、差分脉冲编码调制、标量量化、变长编码和一个梅花形采样的开关。1988 年开发的版本 2 加入了运动补偿和背景预测。由于创建了 ITU-T 来代替以前的 CCITT 标准化团体，1993 年，发表了最终版本。H.120 视频流对于 NTSC（国家电视制式委员会）速率为 1544kbit/s，对于 PAL（逐行倒相制式）速率为 2048kbit/s [S11]。

由于其中的差分 PCM（脉冲编码调制）是逐像素工作的，在实用中 H.120 视频的质量并不好。逐像素工作的差分 PCM 对于空域分辨率很好，但时域方面质量的确很差。因而有必要在不超过视频流的目标比特率的前提下改善视频质量。研究者们提出了 H.120 之后的基于块的编译码方案，例如 H.261 [S4]。

2.4.2 H.261

H.261 [S4，B3] 是首个在实用中获得广泛成功的视频编译码方案（从数量显著的产品支持方面来看）。该 ITU-T 视频编码标准于 1988 年首次设计，是 H.26x 家族的第一个成员。H.261 最初设计是用在 ISDN（综合业务数字网）的线路传输，该线路的数据率是 64kbit/s 的整数倍。它的编码算法综合了运动补偿的帧间预测和 16×16 的宏块运动补偿的空域变换编码，8×8 的 DCT（离散余弦变换）[B2]，标量量化，Z 形扫描和变长编码。所有后来的国际视频编

码标准都深深地植根于 H. 261 的设计［S11］。图 2. 3 所示为 H. 261 编译码方案的概要框图。

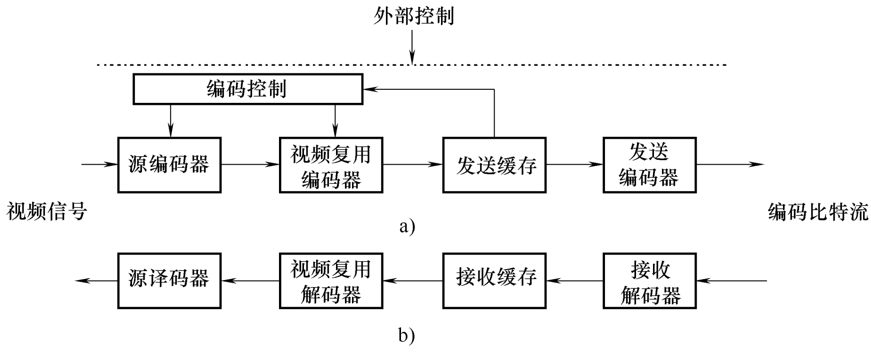


图 2. 3　H. 261 编码器和解码器的概要框图［S4］（ITU-T 1993）
a）编码器　b）解码器

2. 4. 3　MPEG-1

MPEG-1（运动图像专家组）［S3］于 1993 年由 ISO/IEC JTC1 SC29 WG11（MPEG）开发。MPEG-1 为 NTSC 提供 352 × 240（源输入格式）的分辨率，在 1. 5Mbit/s 上为 PAL 提供 352 × 288 的分辨率。当运行在较高比特率上时，MPEG-1 与 H. 261 相比具有更优越的视频质量，且接近于 VHS 的质量。它的主要应用集中在多媒体视频存储上（例如 CD-ROM）。

2. 4. 4　H. 262/MPEG-2

H. 262/MPEG-2 编码标准由 ITU-T 视频编码专家组和 ISO/IEC 运动图像专家组于 1994 年联合开发［S1］。MPEG-2 视频类似于 MPEG-1，但也提供对隔行扫描视频的支持（模拟广播电视系统使用的格式）。MPEG-2 视频在低比特率（低于 1Mbit/s）上没有优化，但在 3Mbit/s 及以上性能超越了 MPEG-1。为保持标准的一致性，MPEG-2 也与 MPEG-1 兼容，这意味着 MPEG-2 播放器可以不加任何修改地播放 MPEG-1 视频。

2. 4. 5　H. 263，H. 263 + 和 H. 263 ++

下一代视频编码超越了 H. 261 成为最主要的视频会议编译码系统。H. 263［S6］在所有比特率上与其以前的标准相比都具有优越的翻倍的视频质量。H. 263 版本 1 由 ITU-T 制订于 1995 年。优于 H. 261［S11］的特征如下：

- DCT 系数的 3-D 变长编码；
- 运动矢量中值预测；
- 双向预测；

- 算术熵编码。

H. 263 + 或版本 2 于 1997 年底 1998 年初制订 [S7]，它包含了许多新特征，如抗误码、自定义灵活的视频格式、增补的增强信息、也存在超越 H. 263v1 的压缩效率。H. 263 ++ 或版本 3 [S6]，制订于 2000 年，在图像质量、包损失和抗误码方面有明显的改善，而且附带了增补的增强信息。

2. 4. 6 MPEG-4

MPEG-4 [S9]，室友 MPEG（运动图像专家组）于 1998 年底制定的 ISO/IEC 标准。为了获得 2000 年初的正式国际标准地位，MPEG-4 版本 2 名下的完全后向兼容的扩展部分于 1999 年底被冻结。为了满足从低质量低分辨率的监控摄像机到高清电视广播和 DVD 的需要，MPEG-4 第 2 部分大约有 21 个档次。部分档次罗列如下 [S9]：

- 简单档次；
- 简单可扩展档次；
- 主档次；
- 核心档次；
- N 比特档次；
- 混合档次；
- 基本动画纹理档次；
- 可扩展纹理档次；
- 简单人脸动画档次；
- 核心可扩展档次；
- 高级可扩展纹理档次；
- 简单 FBA；
- 高级编码效率档次；
- 高级实时简单档次。

2. 4. 7 H. 264/MPEG-4 第 10 部分/AVC

1998 年，ITU-T 视频编码专家组（VCEG）着手一项需要长期努力的工作来起草 "H. 26L" 标准，它将能提供明显好于以往 ITU-T 标准的视频压缩效率。2001 年，ISO 运动图像专家组（MPEG）认可了 H. 26L 的潜在价值，组建了包括来自 MPEG 和 VCEG 专家的联合视频团队（JVT），该团队具有形成新视频编码标准 H. 264/AVC 的许可证。该新标准的正式名称是高级视频编码（AVC）；然而，它的旧的工作名称 H. 26L 广为人知，按照 ITU 的文档编号是 H. 264 [H44，H23，H25，H2]。

图 2. 4 所示为国际视频编码标准年谱（承蒙美国加利福尼亚州圣塔 克拉拉大学计算机工程系的圣菲利波家族讲席教授，Nam Ling 博士）。

与之前的视频编码标准相比，H.264［H44］带来了压缩比的显著提高，也节约了高达 50% 的比特率。通过支持编码的灵活性以及编码数据的组织，该标准能提高抗误码能力。与其他标准相比，H.264 编码效率的提高以及具有的编码灵活性是以提高复杂度为代价的。第 4 章将非常详细地讨论这些特征。

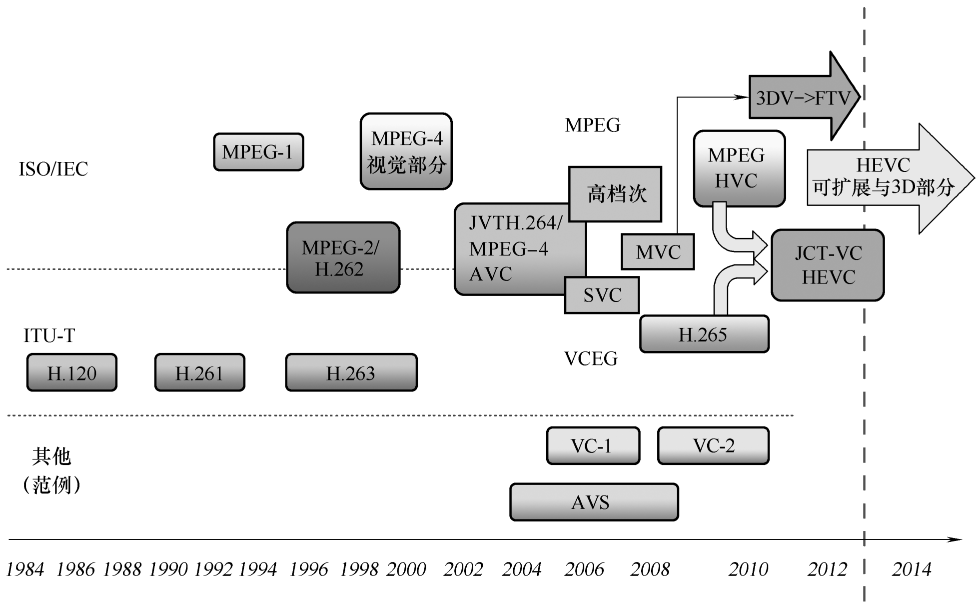


图 2.4　国际视频编码标准年谱

2.4.8　H.265/HEVC

高效视频编码（HEVC）是由 ITU-T/ISO-IEC 正在制订的最新视频编码标准。2013 年 1 月，三个档次（Main、Main10 和 Main 静止图像——只是帧内模式）已被 ITU-T 批准通过。第 5 章将详细说明。

2.5　视频格式和质量

真实世界的典型场景由多个对象、以及它们各自的特征形状、深度、纹理和亮度构成。典型自然视频场景的空间特征和时间特征对于视频处理和压缩都是有关的。空间特征包括场景内的纹理变化、对象的数量和形状、颜色等；时间特征包括对象的运动、亮度的改变、摄像机或视点的运动等。一个自然视觉场景在空间上和时间上都是连续的。把一个视觉场景表示为数字形式包括实际场景的空间和时间采样，空间采样通常在视频图像平面的矩形网格上进行；而时间采样是在固定的时间间隔上成为一系列静止帧或者帧的分量（见图 2.5）。数字视频是采样的视频场景

以数字形式的表示［B18］。

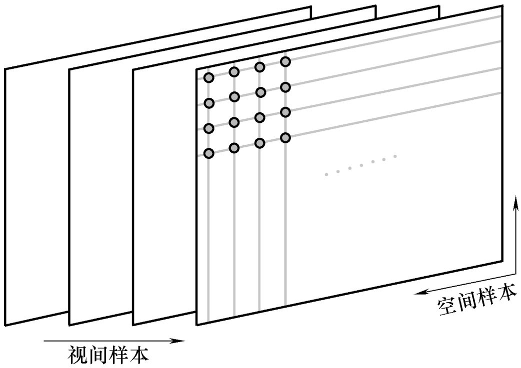


图 2.5 视频序列的空间和视觉采样（2010 Wiley）

2.5.1 帧与场

视频信号可按完整的帧序列渐进地采样，或者是被隔行采样为隔行场序列。在交错视频序列中，两个场构成一幅视频帧（见图 2.6），一个场包括或者是一个完整视频帧的奇数行或者是偶数行。这种采样方法的优点是，在相同的数据率下，它在每秒钟发送的场数可以是相同渐进序列帧数的两倍，同时能给出更平滑的运动视觉效果［B18］。

2.5.2 颜色空间

几乎目前所有的数字视频应用都要显示彩色；因此有必要表示这种压缩信息。颜色空间是表现亮度和颜色的有用方法（见图 2.7）。

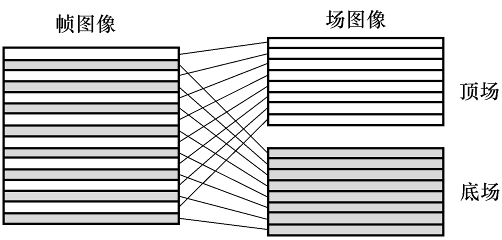


图 2.6 隔行视频序列

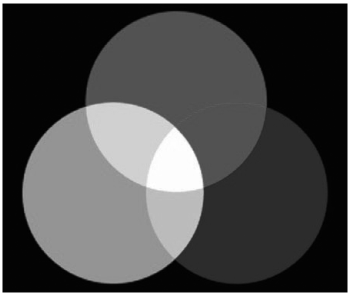


图 2.7 红绿蓝颜色空间

在 RGB 颜色空间中，彩色图像的一个像素由三个数来表示，这三个数指明红、绿和蓝（白光的三个主要合成颜色）的相对比例。任何颜色都可以由不同的比例的红、绿和蓝组合而成。在 RGB 颜色空间中，这三种颜色同等重要，因而通常以

相同的分辨率进行存储。然而，人类视觉系统对于色差的敏感度低于对亮度的敏感度。因此，人们使用了著名的 YUV 颜色空间，通过把亮度和颜色信息分离开来而且以高于颜色的分辨率来表示亮度，YUV 可以更加有效地表示一幅彩色图像。Y 是亮度分量，可被计算为 R、G 和 B 的一种加权平均。

$$Y = k_r R + k_g G + k_b B \tag{2.1}$$

其中，$k_r + k_g + k_b = 1$.

色差信息（色度）可写为

$$Cb = B - Y \tag{2.2}$$

$$Cr = R - Y \tag{2.3}$$

$$Cg = G - Y \tag{2.4}$$

实际上，视频编码只需要传输三个分量（Y、Cb 和 Cr），因为 Cg 可以从 Y、Cb 和 Cr 中得到。按照 ITU-R 的推荐［S12］，$k_r = 0.299$，$k_g = 0.587$ 和 $k_b = 0.114$。那么式（2.2）~式（2.4）可重写为

$$Y = 0.299R + 0.587G + 0.114B \tag{2.5}$$

$$Cb = 0.564(B - Y) \tag{2.6}$$

$$Cr = 0.713(R - Y) \tag{2.7}$$

$$R = Y + 1.402Cr \tag{2.8}$$

$$G = Y - 0.344Cb - 0.714Cr \tag{2.9}$$

$$B = Y + 1.772Cb \tag{2.10}$$

图 2.8 所示为一幅彩色图像的红、绿和蓝分量，图 2.8b 所示为相应的色度分量 Cr、Cg 和 Cb 分量以进行对比。



a)



b)

图 2.8　a）彩色图像的红、绿和蓝分量［B18］（2010 Wiley）
b）彩色图像的 Cr、Cg 和 Cb 分量［B18］（2010 Wiley）

YCbCr 采样格式

图 2.9 所示为现代视频编码标准如 MPEG-4 视觉和 H.264 支持的 Y、Cb 和 Cr 的三种采样模式。4:4:4 采样保持了色度分量的完整信息。三个分量 Y、Cb 和 Cr 具有同样的分辨率，即对于每四个亮度样值就有四个 Cb 和 Cr 样值。4:2:2 采样也称为 YUV2，色度分量与亮度一样具有同样的垂直分辨率，但水平分辨率为后者的一半。也就是说，对应水平方向的每四个亮度样值有两个 Cb 和两个 Cr 样值。4:2:2 视频通常用于高质量颜色再生。

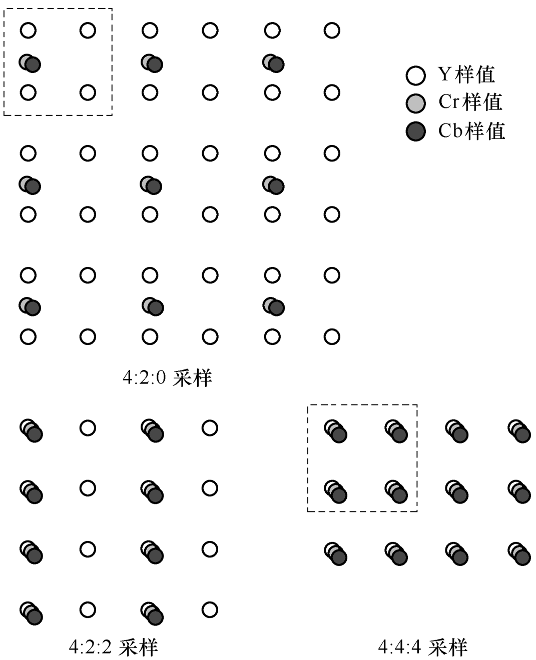


图 2.9 4:2:0，4:2:2 和 4:4:4 的采样模式（渐进式）([B18] 2010 Wiley)

最流行的采样模式是 4:2:0，也叫作 YV12。Cb 和 Cr 在水平和垂直方向上的分辨率都是 Y 分辨率的一半，每个色差分量包括 Y 分量样值数量的四分之一。4:2:0 YCbCr 视频只需要4:4:4或者 RGB 视频样值的一半，因此被广泛用于消费应用领域诸如视频会议、数字电视和 DVD [B18]。

2.5.3 视频格式

一种非常普通的做法是，在压缩和传输之前捕获或转换视频为中间格式的一种。表 2.1 示出了部分流行的视频格式。

帧分辨率的选择取决于具体应用和可用的存储或传输容量。例如，4CIF 适合于标准清晰度电视和 DVD 视频；CIF 和 QCIF 则流行于视频会议应用领域；QCIF 或 SQCIF 适合于显示分辨率和比特率有限的移动多媒体应用领域。

SIF（Source Input Format）实际上等同于 CIF，但来自于 MPEG-1 而不是 ITU 标准。在 525 行的 NTSC 系统上的 SIF 分辨率是 352×240，在 625 行 PAL 系统上的 SIF 等同于 CIF，分辨率是 352×288。SIF 和 4SIF 通常用在特定的视频会议系统中［H53］。

表 2.1　视频帧格式

格　　式	视频分辨率
Sub-QCIF	128×96
Quarter CIF（QCIF）	176×144
SIF（525）	352×240
CIF/SIF（625）	352×288
4SIF（525）	704×480
4CIF/4SIF（625）	704×576
16 CIF	1408×1152
DCIF	528×384

2.5.4　质量

为了规定、计算和评价的便利，有必要确定呈现给观察者的视频图像的质量。视频质量本身是个主观概念，它受到许多因素的影响，因而获得完全精确的质量量度是很困难的。用客观标准度量视觉质量能给出精确的可重复的结果，但迄今为止，还没有哪一个客观度量体制能完全地再生一个视频观察者的主观经验［B18］。

1. 峰值信噪比

峰值信噪比（PSNR）是最为广泛使用的客观质量指标。PSNR［方程(2.11)］以对数来计算，取决于原始图像和损失图像或视频帧之间的均方误差(MSE）与（$2^n-1)^2$ 的相对值（图像中可能的最大信号幅值的平方，n 是每个图像样值的比特数）。

$$PSNR_{dB}=10\log_{10}((2^n-1)^2/MSE) \tag{2.11}$$

PSNR 易于快速计算，因而是一个非常流行的质量指标，被广泛用于比较压缩和解压视频图像的质量。

2. SSIM

结构相似度（SSIM）指标［Q13］（见附录 C）是一种度量两幅图像之间相似度的方法。如果另一幅图像被视为具有完美的质量，SSIM 可被看作是被比较图像的一种质量指标。

2.6 总结

本章基本上是第 1 章的延续。下一章将详细描述 AVS China。

附言：本章主要基于 S. M. Muniyappa 的学位论文“帧内模式选择复杂度降低算法的实现”（“Implementation of complexity reduction algorithm for intra mode selection”），2011 年 12 月，德克萨斯大学阿灵顿分校电气工程系。该论文可以通过网址 www-ee. uta. edu/dip 访问。点击课程然后点击 EE5359，再下拉即可看到 Theses/Projects。

第 3 章　AVS China

摘要：详细地描述了中国的音频和视频标准（AVS），说明了编码器和解码器固有的档次、级别和功能特性。该编译码系统类似于 H. 264/AVC（第 4 章），但选项更加简单，所以降低了实现复杂度，而性能损失可以忽略。此外，也呈现了 AVS 系统的细节和 AVS China 采纳为 IEEE 标准的情况。

关键字：AVS China　档次　数据格式　AVS 编码器　解码器　编码工具　比特流　NAL 单元　AVS-M　IEEE AVS

3.1　AVS China

AVS（Audio Video coding Standard：音频视频编码标准）China 是由中国的 AVS 工作组制订的最新的数字视频编码标准，以降低因使用其他国际视频编码标准如 MPEG-2、MPEG-4 和 MPEG-4 第 10 部分（H. 264）[A14] 而需要支付的版税费用。AVS China 的主要特性是，在技术上它是一种先进的第二代信源编码标准，而且完全由中国人规划和控制 [AS1]。AVS 标准是一个高效的视频编码器，它使用了最新的视频编码工具，主要瞄准标清（SD，Standard Definition）和高清（HD，High Definition）视频压缩，旨在实现与 H. 264/AVC 类似的编码效率但是具有更低的计算复杂度 [PC2]。AVS 视频标准制订的目的是服务于广播和存储媒体应用如数字视频电视、数字视频光盘（DVD 和高清光盘）以及宽度网络多媒体应用如视频会议、视频点播、IPTV 等。因为 AVS China 的视频编码语法结构非常类似与 MPEG-2 视频标准，所以它能很容易地用在目前广泛使用的 MPEG-2 系统中，同时在编码效率方面有明显的提高 [A6]。AVS 视频编码标准是 AVS 工作组标准化产品的重要组成部分。AVS-video 是所有与视频编码相关部分的综合体，它的辅助信息在 AVS 之内。不同的 AVS 部分见表 3.1。

表 3.1　AVS 标准的不同部分 [A13]

部　分	名　称
1	系统
2	视频
3	音频
4	一致性测试
5	参考软件
6	数字版权管理

（续）

部　　分	名　　称
7	移动视频
8	通过 IP 网络传输 AVS
9	AVS 文件格式
10	移动语音和音频编码

AVS China 的第二部分，也就是部分 2（视频）主要服务于高清和高质量的数字广播应用、数字存储媒体和其他相关应用，而 AVS 的第 7 部分主要服务于移动多媒体应用方面。AVS 第 2 部分的视频编码器结构非常类似于 H. 264 标准，但是它的复杂度通过只选择 8 ×8 的块、五种帧内模式和其他特征而得以降低［G7］。

各种视频编码标准的逐年发展如图 3. 1 所示。图 3. 2 也示出了 AVS 档次及其应用。

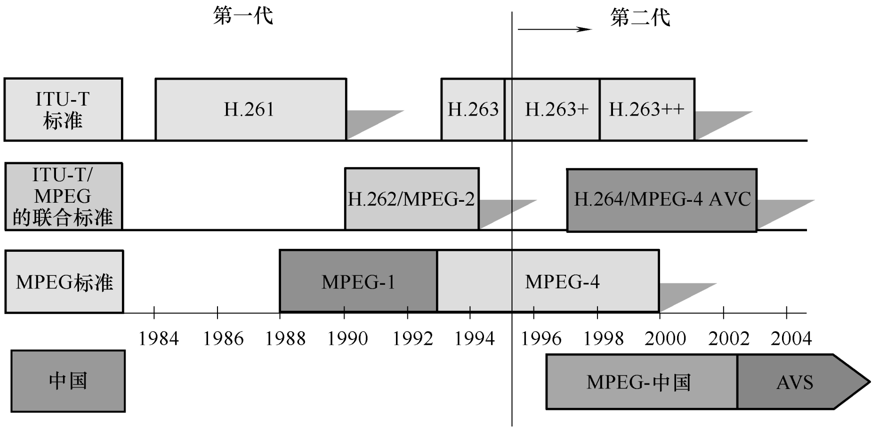


图 3. 1　视频编码标准的历史［A40］

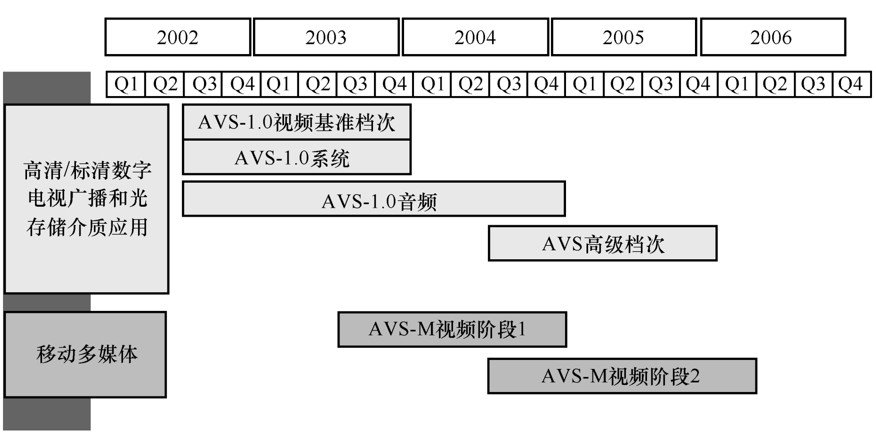


图 3. 2　AVS 档次的标准结构［A40］

3.2 AVS China 档次和级别

考虑到视频应用的不同需求，AVS-video 定义了不同的档次，这些档次综合了先进的视频编码工具，考虑了编码效率和计算复杂度之间的平衡问题，能服务于多种不同的应用。标准定义的基本档次见表 3.2。

AVS-video 定义的 4 个不同档次是

- 基准档次；
- 基本档次；
- 伸展档次；
- 加强档次。

表 3.2 AVS China 档次特征［A59］

档次	基准档次	基本档次	伸展档次	加强档次
可用的颜色格式	4:2:0，4:2:2	4:2:0	4:0:0，4:2:2	4:2:0，4:2:2
最大块单元及变换尺寸	8×8	4×4	8×8	8×8
帧内预测	8×8 帧内预测	4×4 帧内预测	8×8 帧内预测	8×8 帧内预测
帧间预测	P 预测和 B 预测	只有 P 预测，非参考 P	兼有 P 预测和 B 预测，背景参考帧，非参考 P	兼有 P 预测和 B 预测
内插	两步四抽头内插	两步四抽头内插	两步四抽头内插	两步四抽头内插
最大参考帧数	2	2	2	2
量化	固定量化	固定量化	固定量化，加权量化，场景自适应加权量化	固定量化，加权量化，场景自适应加权量化
熵编码	8×8 二维变长码	4×4 二维变长码	8×8 二维变长码	8×8 二维变长码，8×8EAC
对隔行方式的支持	帧编码或场编码	只有帧编码	帧编码或场编码	帧编码，场编码，或者 PAFF
差错恢复		SEI 内的场景信令	核心图像，灵活的图像首部，灵活的分片组，受限的 DC 帧内预测	

3.2.1　AVS-video 的基准档次

基准档次被认为是计算复杂度中等的基础档次，它定义在 AVS 的第二部分，主要服务于像存储媒体和商业广播这样的数字视频应用领域。

3.2.2　AVS-video 基本档次

基本档次是个基础档次，定义在 AVS 第 7 部分，主要服务于移动视频应用。

3.2.3　AVS-video 伸展档次

伸展档次是定义在 AVS 第 2 部分的一个扩展档次，它专注于视频监控应用标准化的解决方案。

3.2.4　AVS-video 加强档次

加强档次定义在 AVS 第 2 部分，主要服务于电影压缩的高密度存储。

AVS-video 定义的多种档次简述及其他们的应用见表 3.3。

档次是 AVS 视频标准所定义的语法、语义和算法的一个子集，而级别则对视频流的参数加以限制 [A10]。AVS 视频标准定义了四个级别。

- 4:2:0 和 4:2:2 格式的标清（SD）视频级别 4.0 和 4.2 [A6]；
- 4:2:0 和 4:2:2 格式的高清（HD）视频级别 6.0 和 6.2 [A6]。

表 3.3　AVS-vide 档次应用 [A59]

档　次	部　分	主要应用
基准档次	2	电视广播，高清电视
增强档次	2	存储媒体 [A40]
基本档次	7	移动应用
伸展档次	2	视频监控
加强档次	2	多媒体娱乐

AVS 视频定义的最大图像尺寸范围是 720×576～1920×1080 像素，最大比特率范围是 10Mbits/s 到 30Mbits/s。

接下来，AVS China 进行了许多改变。部分改变如下：

在 AVS 第 2 部分（见表 3.1），移动编码作为一个新的档次包含在内，也叫作便携档次。AVS 第 7 部分（AVS-M）仍然独立存在，但人们不再太多关注它。目前，档次被称作组群。

主组群对应基准档次。便携组群对应基本档次。监控基本组群对应伸展档次。监控组群是监控基本组群的一个新改进。加强组群对应加强档次。广播组群是一个最近为中国电视广播新制订的档次。

3.3　AVS 使用的数据格式

AVS 对视频数据以渐进扫描格式进行编码，每帧的各行被依次扫描；而隔行扫描对一帧的奇场和偶场进行交替的扫描。渐进扫描格式的显著优势就是运动估计得以有效工作。渐进扫描帧能以明显低于达到相同视觉质量的隔行编码帧的比特率进行编码；而且，渐进格式数据的运动补偿编码复杂度也明显低于隔行数据的编码，隔行数据相当于 AVS 编码中复杂度降低的一个有意义分量［A6］。不过，AVS 对于隔行扫描格式也提供了编码工具。

3.3.1　AVS 视频分层结构

AVS 建立在表示视频数据的分层结构上，如图 3.3 所示。

1. 序列

序列层包括多组连续视频帧，提供了一个编码视频的入口点，包括一组强制和可选的系统参数。强制系统参数对于初始化解码器系统是必需的，而可选系统参数则用于其他的系统设置，由网络提供方自行决定。可选的用户数据也可放在序列首部进行发送。如图 3.4 所示为视频序列的一个例子。

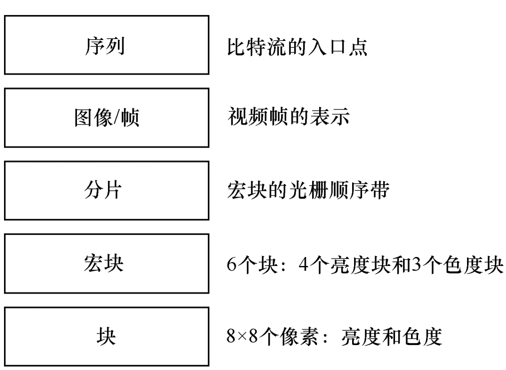


图 3.3　AVS 分层的数据结构

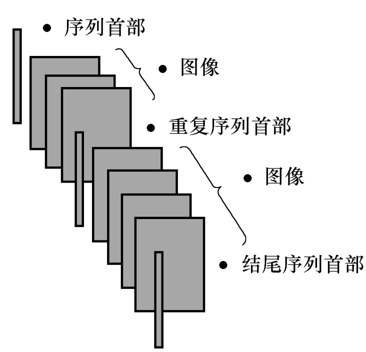


图 3.4　视频序列

2. 图像

图像层提供了视频数据的编码信息，也包括一个带有强制和可选参数的首部，以及可选的用户数据。AVS 定义的三种图像类型如图 3.5 所示。

- 内部编码帧（I 帧）；
- 前向解码帧（P 帧）；
- 预测帧（P）——至多两个参考帧（P 或 I）；
- 双向解码帧（B 帧）；
- 内插帧（B）——两个参考帧（I 或 P 或都有）。

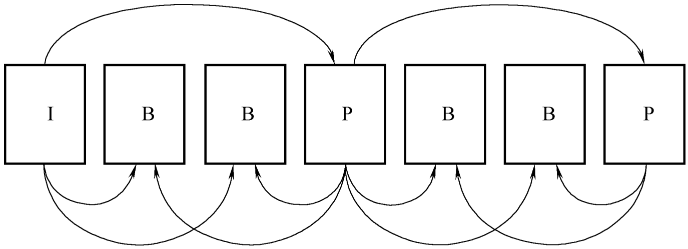


图 3.5　I、P、B 帧格式

在一个视频标准里规定有三种不同顺序的帧［B18］：

♦ 解码顺序是指从比特流中解码图像的顺序；

♦ 显示顺序是指图像显示的顺序；

♦ 参考顺序是指预测其他帧的参考帧安排顺序。

在 AVS-video 标准中，如果比特流不包括编码的 B 帧，那么图像的解码顺序与显示顺序相同；但是当有 B 帧的时候，解码顺序则不同于显示顺序，解码图像应该按照下列规则重新排序［A12］：

• 如果当前解码帧是一个 B 帧，那么当前解码帧直接输出显示；

• 如果当前解码帧是一个 I 帧或 P 帧，那么就传输并显示以前解码的 I 帧或 P 帧，如果存在的话；否则不传输任何图像；

• 当所有图像都已被解码，如果缓冲区仍有解码的图像，就把它们传输到显示器。

AVS 视频标准帧处理过程的图形表示如图 3.6 所示。

编码器输入顺序:

1	2	3	4	5	6	7	8	9	10	11	12	13
I	B	B	P	B	B	P	B	B	I	B	B	P

编码和解码顺序:

1	2	3	4	5	6	7	8	9	10	11	12	13
I	P	B	B	P	B	B	I	B	B	P	B	B

编码器的输出顺序(显示顺序):

1	2	3	4	5	6	7	8	9	10	11	12	13
I	B	B	P	B	B	P	B	B	I	B	B	P

图 3.6　AVS 视频标准的帧处理［A2］

帧组（GOP，Group of pictures）规定了内部编码帧和帧间编码帧安排的顺序，它可以包含下列图像类型：

• I 帧（内部编码帧）：它是一幅独立于其他图像类型的参考图像。每个 GOP

以 I 帧开始。

- P 帧（预测编码帧）：它包含来自前面 I 或 P 帧的运动补偿差分信息。
- B 帧（双向预测编码帧）：它包含来自前面和后面的 I 或 P 帧的运动补偿差分信息。

图 3.6 中帧的重排序可以解释如下。

例如，在连续的 I 和 P 帧之间有两个 B 帧，在连续的 P 帧之间也有两个 B 帧。帧“1I”被帧“4P”用作预测。帧“4P 和“”1I”都用于预测‘2B’和‘3B’。因此，帧的解码顺序为‘1I’，‘1P’，‘2B’‘3B’，显示顺序为‘1I’，‘2B’，‘3B’，‘4P’。

3. 分片

在发生比特传输错误的情况下，分片结构提供重新同步的最底层机制。分片包括数量可变的宏块。

4. 宏块

一个宏块表示一个 16×16 的图像区域，包括亮度分量（Y）和色度分量（Cb 和 Cr）像素。总共有三种 Y，Cb 和 Cr 的采样模式，AVS 视频标准支持其中的两种，它们是 4∶2∶0 和 4∶2∶2 格式［A6］。其中的数字指出水平方向上每个分量的相对采样率。4∶2∶2 格式表示水平方向上对应每 4 个亮度样值有 2 个 Cb 和 Cr 样值，而在 4∶2∶0 的采样格式中，Cb 和 Cr 各自有 Y 的垂直和水平分辨率的一半。宏块格式的图形表示如图 3.7 和图 3.8 所示。

在图 3.7 所示的 4∶2∶0 采样格式中，每个色度分量，即 Cr（8×8）和 Cb（8 × 8）分量的垂直和水平分辨率是 Y（16×16）分量的一半。

在图 3.8 所示的 4∶2∶2 采样格式中，对于水平方向上每 4 个亮度样值，就有 2 个 Cb和 Cr 的样值。

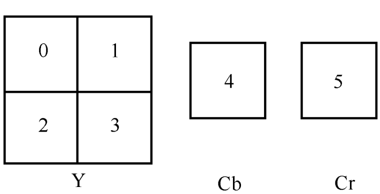


图 3.7　4∶2∶0 宏块格式

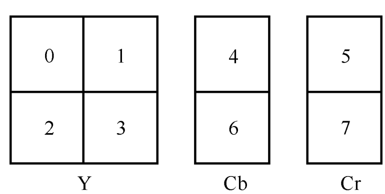


图 3.8　4∶2∶2 宏块格式

5. 块

块是最小的编码单元，它包含预测误差的变换系数数据。

3.4　AVS 视频编码器

AVS 视频的基本编码过程如图 3.9 所示。

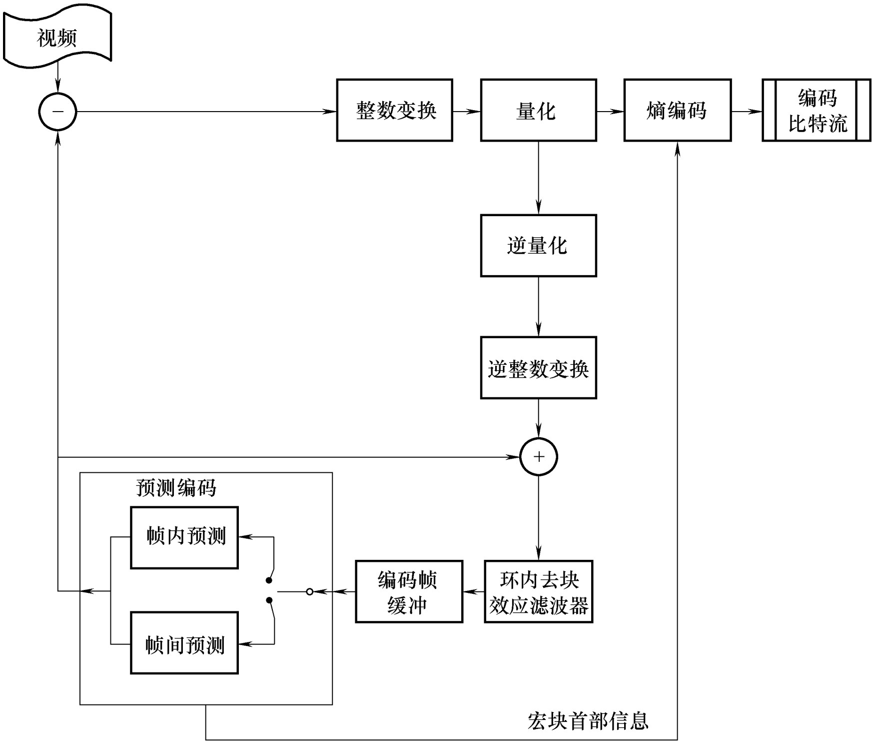


图 3.9 AVS 视频编码器的框图［A59］（2009 Elsevier）

类似于以往的视频编码标准如 MPEG-1、MPEG-2 和 H.264［B18］，AVS 第 2 部分也使用了基于混合块的视频压缩技术如空间和时间预测、变换、量化、熵编码和去块效应滤波器来获得编码效率和复杂度之间较好的平衡［PC2］。通过运动补偿 DPCM 编码去除时间冗余；首先通过空间预测，最后由变换编码来去除残差的空间冗余；通过熵编码去除统计冗余。

3.4.1 编码过程概述

一个视频包括一个帧的序列（YUV），每一帧都要分成若干叫作宏块的矩形块，宏块包括固定大小的 16×16 亮度分量及其相应的色度分量。在每个宏块上进行预测编码，预测编码可被分为或者是帧内编码或者是帧间编码两类。然后，对相应于预测残差的宏块进行变换，残差是当前图像原始像素值和和预测像素值之间的差值。在进行熵编码之前，变换系数要进一步被量化和扫描（Z 形）（见后面的图 3.12），最终熵编码信息被转换成一个比特流［A59］。

3.4.2 AVS 视频编码器用到的编码工具

AVS 视频编码器所用的各种编码工具的功能描述如下。

1. 变换

与 H. 264 和 MPEG-2 不同，AVS 使用的是可分离整数精度的 8×8 离散余弦变换。该变换与量化结合起来设计来最小化解码器的实现复杂度，该变换叫作预缩放的整数余弦变换技术（PIT）[I13]。图 3.9 和图 3.10 分别示出了 H. 264 所用的传统 ICT 方案和 AVS 所用的 PIT 方案的框图（见图 3.11）。

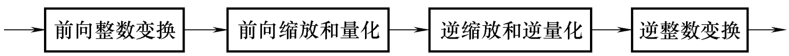


图 3.10　H. 264 使用的传统的 ICT 方案框图 [I13]（2005 IEEE）

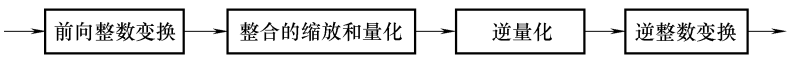


图 3.11　AVS-video 使用的 PIT 方案框图 [I13]（2005 IEEE）

一个 8×8 逆变换矩阵如下：

$$[H]=\begin{bmatrix} 8 & 10 & 10 & 9 & 8 & 6 & 4 & 2 \\ 8 & 9 & 4 & -2 & -8 & -10 & -10 & -6 \\ 8 & 6 & -4 & -10 & -8 & 2 & 10 & 9 \\ 8 & 2 & -10 & -6 & 8 & -9 & -4 & -10 \\ 8 & -2 & -10 & 6 & 8 & 9 & -4 & 10 \\ 8 & -6 & -4 & 10 & -8 & -2 & 10 & -9 \\ 8 & -9 & 4 & 2 & -8 & 10 & -10 & 6 \\ 8 & -10 & 10 & -9 & 8 & -6 & 4 & -2 \end{bmatrix} \tag{3.1}$$

假定 $[Y]$ 是一个变换系数矩阵，逆变换共包括如下 4 个步骤 [A6]：

A. $[W_1]=[Y][H]^{\mathrm{T}}$，其中 $[W_1]$ 是 1-D 逆变换的结果。

B. $[W_2]=([W_1]+2^2)\gg 3$，这意味着 $[W_1]$ 的每个元素除以 8 并取整。

C. $[W_3]=[H][W_2]$，其中 $[W_3]$ 是 2-D 逆变换之后的结果。

D. $[W]=([W_3]+2^6)\gg 7$。

2. 量化与扫描

对于渐进数据进行 Z 形（见图 3.12）扫描而对于交错数据进行隔行扫描，这样就把由变换编码生成的 2D 系数要转化成 1D 序列以进行量化和编码。图 3.12 所示为用于渐进数据的 Z 形扫描。

AVS 采纳了自适应均匀量化器对变换系数进行量化。AVS-video 不提供改变比特率的直接选项，因为可以改变量化器的步长来实现率控制，率控制在避免缓冲溢出的恒定比特率工作中是很有用的。量化参数对于一整幅图像或分片可以是固定的，或者在每个宏块上可以有差别地更新。

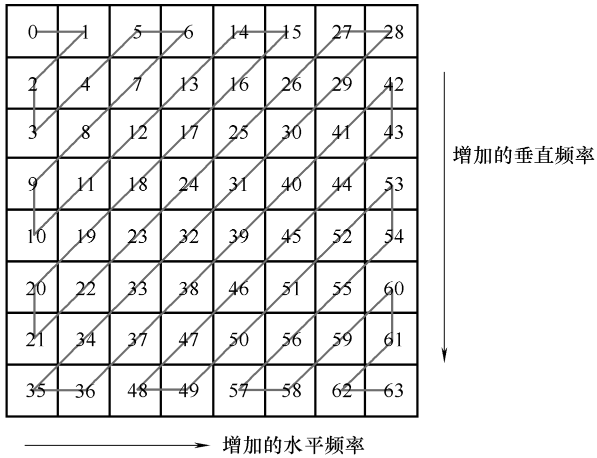


图 3.12 AVS 中用于渐进数据的 Z 形扫描

3. 熵编码

熵编码在视频压缩中减少 DCT 系数的统计相关性方面发挥着重要的作用。AVS 使用了一个基于上下文的 2D-VLC（C2DVLC）熵编码器。C2DVLC 用到多个简单指数 Golomb 码的 VLC 表，利用基于上下文的自适应多表编码来开发每个块 DCT 系数的统计相关性。这种相关性的出现是因为，随着 DCT 子带频率的增大，非零系数的幅值减小而连续的零系数游程变长。使用指数 Golomb 码对零游程和非零系数进行编码可获得较高的编码效率。就编码效率而言，与 H.264/AVC 所用的基于上下文的自适应变长码（CAVLC）相比，C2DVLC 具有类似的编码效率但计算复杂度更低 [A23]。

4. 去块效应滤波器

去块效应滤波器在运动补偿和残差编码之后作用于宏块通过平滑宏块周围的边沿来改善视觉质量和预测性能。环路去块效应滤波器用来降低块效应，因而提高视觉质量。AVS-video 定义了几个去块效应方法，这些方法随滤波器的强度而有所不同，滤波器强度取决于相邻块的编码模式、量化步长和块间亮度梯度的陡度 [A59]。

- 默认的去块效应滤波器运行在 8×8 的边界上（见图 3.13）；
- 简化的去块效应滤波器运行在 4×4 的边界上；
- 像素级环路去块效应滤波器在像素级别而不是在边界级别上决定了边界强度（BS：boundary strength）。

默认的去块效应滤波器运行在 8×8 的块边界上。首先，对垂直边界进行水平滤波，然后对水平边界进行垂直滤波。以 4∶2∶0 图像为例，宏块边界需要按图 3.13所示的实线（垂直边界）和虚线（水平边界）进行滤波，其中每个方框代

表一个 8×8 的块，点线表示后面要与相邻宏块处理的边界。参照预测类型或临近块边界的两个相邻块的参考序号，在默认的去块效应滤波器中设计了三级 BS。在滤波过程中边界每一侧的三个像素（见图 3.14）用来调整边界的像素，如下所示。如果临近当前块边界的两个块至少有一个是内部编码宏块，那么 BS 设置为 2。否则，如果当前块边界的两个相邻块具有相同的参考帧而且它们的运动矢量之差小于一个像素，BS 设置为 0。否则，BS 设置为 1。当处理一个特定块边界时，对于 BS＝1 和 2，进行滤波过程；对于 BS＝0，不使用滤波器。

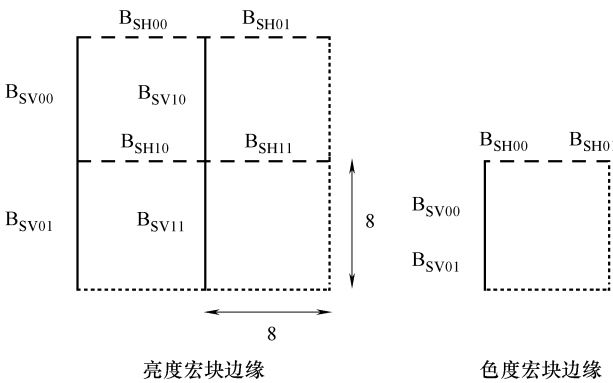


图 3.13　要滤波的一个宏块的块边沿（4∶2∶0 格式）[A2]

p_2	p_1	p_0	q_0	q_1	q_2

图　3.14

5. 模式判决

模式判决单元为图像和宏块选择最好的运动补偿模式。率失真优化用来提高模式判决的效率。

6. 帧内预测

帧内预测是种预测方法，用来利用当前帧的空间相关性。它使用当前帧的已解码信息作为预测的参考来消除同一帧像素间的空间相关性。AVS 第 2 部分所用到的 8×8 内部预测方法允许针对亮度分量的五种预测模式，即 DC（模式 2）、水平（模式 1）、垂直（模式 0）、左下（模式 3）和右下（模式 4），和针对色度分量的四种预测模式，即 DC、水平、垂直和平面。四个 8×8 亮度块的每一个可通过选择五个不同预测模式中最好者进行预测。在使用 DC、对角左下和对角右下模式之前，要用一个三抽头低通滤波器（1/4，2/4，1/4）作用于样值，这些样值将被用作预测的参考［A59，PC2］。(见图 3.15～图 3.20)。

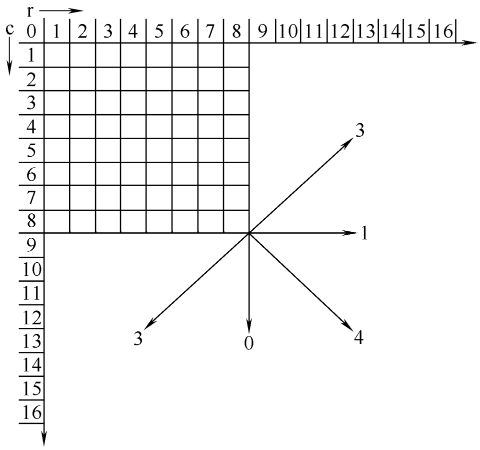


图 3. 15　亮度分量 8 × 8 帧内预测所用的方向和参考像素（2005 SPIE）

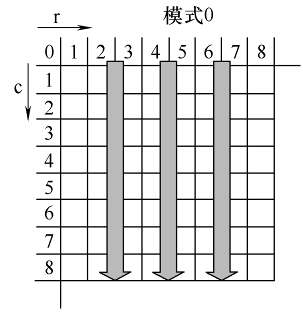


图 3. 16　垂直模式（模式 0）（2005 SPIE）

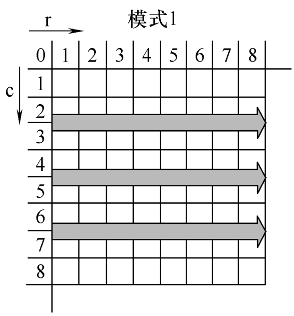


图 3. 17　水平模式（模式 1）（2005 SPIE）

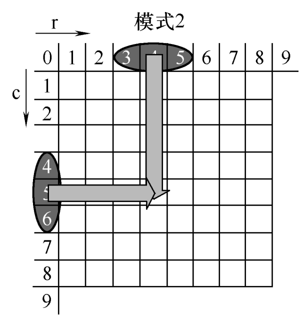


图 3. 18　DC 模式（模式 2）（2005 SPIE）

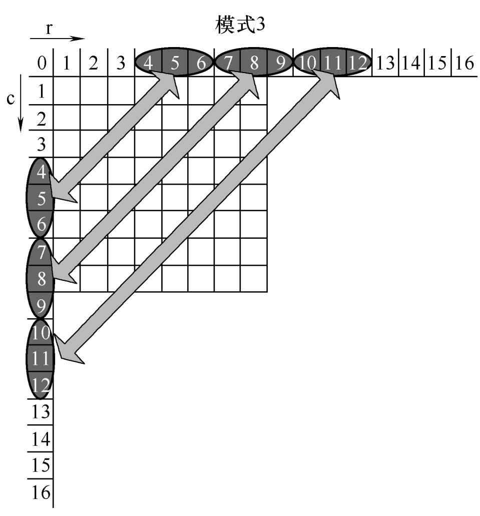


图 3. 19　对角左下模式（模式 3）[A14]（2005 SPIE）

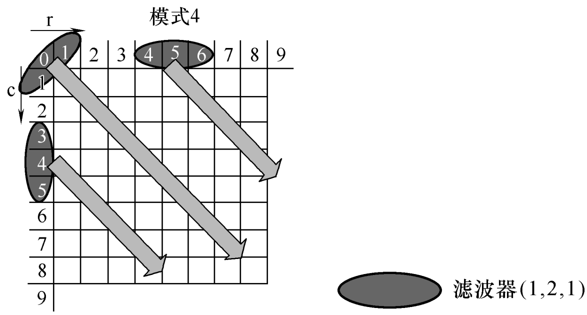


图 3.20　对角右下模式（模式 4）[A14]（2005 SPIE）

• 在垂直模式（模式 0）中，上面的样值由垂直外插得到；

• 在水平模式（模式 1）中，左边的样值由水平外插得到；

• 在直流模式（模式 2）中，所有的样值由他们对应的左边和上面的样值均值来预测；

• 在对角左下模式（模式 3）中，样值在左下和右上之间的 45°角度上内插得到；

• 在对角右下模式（模式 4）中，样值在从下到右的 45°角度上外插得到。

7. 帧间预测

帧间预测是又一种预测方法，它开发了不同帧间的时间相关性。帧间预测涉及两个重要的概念，叫作运动估计（ME，Motion Estimation）和运动补偿（MC，Motion Compensation）。

首先将一个帧间编码帧分为若干宏块。编码器在先前已编码帧（参考帧）里搜索一个类似的块，而不是直接对原始像素值进行编码。如果编码器成功地找到了匹配块，那么当前宏块就用一个指向参考帧中匹配宏块位置的运动矢量来编码。这个确定运动矢量的过程叫作运动估计。被选的宏块区域就是当前宏块的预测值，从当前宏块减去该预测值，形成一个残差块。形成残差块的过程叫作运动补偿。残差块和差分运动矢量一起被编码和传输。解码器接收运动矢量并用它们生成预测区域，预测区域进一步叠加到残差块上重建成原始的宏块（见图 3.21）。

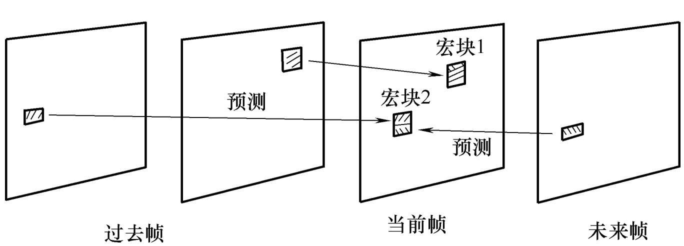


图 3.21　帧间预测

AVS 第 2 部分在帧间预测中支持可变块大小的运动补偿，块大小可以是 16 × 16 或 8 × 8 以更好地利用时间相关性。AVS 第 2 部分所用的变块大小运动补偿如图 3. 22 所示。

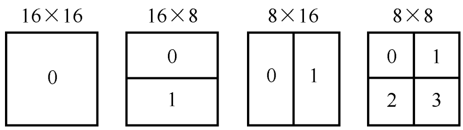


图 3. 22　AVS 第 2 部分帧间预测所支持的块大小［PC2］（2006 Springer）

AVS 第 2 部分支持 1/4 样值精度的运动补偿。AVS 第 2 部分使用了 1/4 像素的二维可分离内插法，叫作两步四抽头内插（TSFT，Two Steps Four Taps Interpolation）［PC2］。分数样值内插用在一个对象在帧间移动了非整数个像素的情况，如图 3. 23所示。

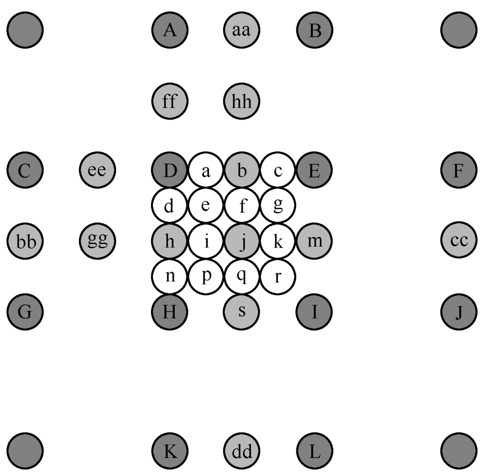


图 3. 23　AVS 第 2 部分所用的 1/4 像素精度的样值（2005 SPIE）

通过在整数像素位置上的样值使用一个四抽头三次卷积内插滤波器$\left(\text{抽头值为} -\frac{1}{8},\ \frac{5}{8},\ \frac{5}{8},\ -\frac{1}{8}\right)$，内插得到半像素位置 b，h 上的样值，位置 j 上的值通过在半像素位置上进行内插获得。1/4 像素位置 a，c，d，f，i，k，n 和 q 上的样值由四抽头三次样条滤波器$\left(\text{抽头值为}\frac{1}{16},\ \frac{7}{16},\ \frac{7}{16},\ \frac{1}{16}\right)$在整数和半像素位置上内插得到。1/4 像素位置 e，g，p 和 r 上的样值由双线性滤波器在半像素位置 j 和整数像素位置即 D，E，H 和 I 上的值内插得到［PC2］。

AVS 第 2 部分也支持多参考帧对 P 和 B 帧进行运动补偿，但是 P 或 B 帧所用的参考帧数限制为 2 以降低存储需求和计算复杂度。P 帧至多使用两个前面的帧进

行帧间预测，而 B 帧至多允许不超过一个前向帧和一个后向帧用于帧间预测［PC2］。

3.5　AVS 视频解码器

AVS 视频解码器的框图如图 3.24 所示。

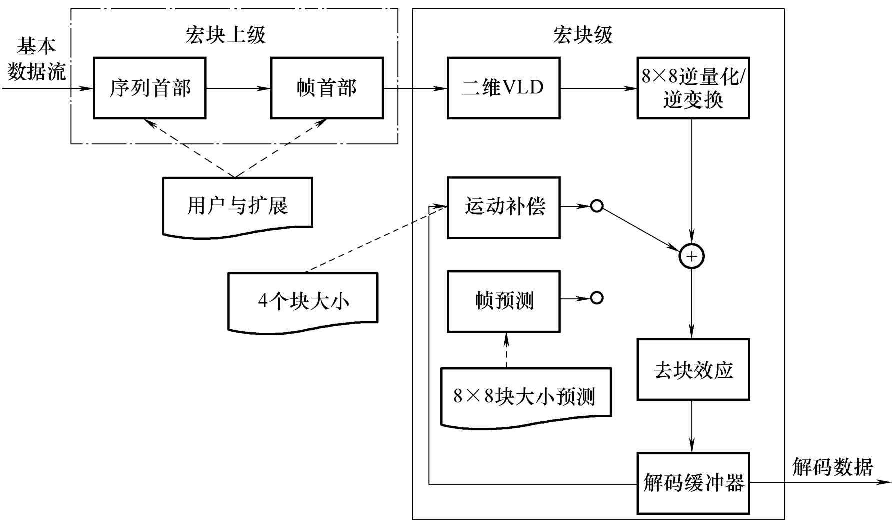


图 3.24　AVS 视频解码器［A31］（2006 IEEE）

AVS 解码器从存储媒体或传输介质中接收基本的压缩视频流作为其输入并将其存储在一个速率缓冲器中，数据以每个宏块和帧所需的速率从该缓冲器中读取出来。比特流解析器紧随其后，将量化参数、运动矢量和其他边信息从编码数据中分离出来。该数据再经过 VLD 熵解码器提取出首部信息和分片数据以及运动矢量。然后，该信号有逆量化器和逆 DCT 进行解码以重建预测误差或者编码数据。运动矢量有运动补偿单元解码来生成当前帧的预测帧，该预测值进一步加到预测误差上来生成输出信号。

3.6　AVS 视频比特流

AVS 视频比特流所用的部分语法和语义解释如下。

3.6.1　起始码

起始码是用在 AVS 视频流中的一种特殊的比特模式。每个起始码包括一个起始码前缀和一个起始码值。起始码前缀是一个 23 个零比特后接一个值为 1 的比特，

即比特字符串“0000 0000 0000 0000 0000 0001”或“0x000001”，且所有的起始码按字节对齐。在 C/C++/Java 编程语言中，十六进制数冠以 ox。

3.6.2　起始码值

每个起始码前缀后接一个起始码值，是一个指明起始码类型的 8 比特整数。表 3.4 描述了 AVS 视频比特流用到的各种起始码类型及其值。

1. 视频编辑码

使用的比特字符串是“0x000001B7”。该语法意思是，对于紧跟一个 I 帧的连续 P 帧或 B 帧可能有漏掉的参考帧，而且这个 P 或 B 帧不能被直接编码。

2. 视频序列终止码

字符串是“0x000001B1”。该语法指明视频序列的结尾。

3. 视频序列起始码

字符串是“0x000001B0”，该语法指明一个视频序列的开始。

表 3.4　AVS 使用的起始码类型和起始码值［A2］

起始码类型	起始码值（十六进制数）
分片起始码（I、P 或 B）	00 ~ AF
视频序列起始码	B0
视频序列终止码	B1
用户数据起始码	B2
I 帧起始码	B3
保留	B4
扩展起始码	B5
pb 帧起始码	B6
视频编辑码	B7
保留	B8
系统起始码	B9 ~ FF

4. 扩展起始码

比特串为“0x000001B5”，指明扩展的开始。

5. 用户数据起始码

比特串为“0x000001B2”，指明用户数据的开始。用户数据被连续存储直到下一个起始码。

6. I 帧起始码

比特串为“0x000001B3”，该语法是指明 I 帧的开始。

7. PB 帧起始码

比特串为“0x000001B6”，该语法是指明 P 或 B 帧的起始码。

8. 分片起始码

比特串包括总共 32bit，就像其他起始码一样。前 24bit 的值为“0x000001”，后 8bit 的范围从 0x00 到 0xAF，是的分片垂直位置。分片垂直范围给出宏块单元中的分片里第一个宏块的垂直位置。

3.6.3　帧编码类型

用于 AVS 视频比特流的另一个重要的语法元素是图像编码类型。该语法是个 2bit 的无符号数，规定了图像编码的类型。语法结构见表 3.5 所示。

表 3.5　P 或 B 帧的编码类型

图像编码类型	编 码 类 型
00	禁用
01	前向帧间预测（P）
02	双向帧间预测（B）
03	保留

3.7　AVS 视频流的 NAL 单元

NAL（network abstraction layer）单元代表网络抽象层单元。它是一种信息分包层类型，给编码的视频比特流加上某种前缀。设计 NAL 单元的主要目的是为视频数据的传输提供一种友好的网络环境。它主要处理视频相关的应用诸如视频电话、视频存储、广播和流媒体应用，IPTV 等。H. 264/AVC 标准［H3］定义了 NAL 单元的语法但 AVS 标准第 2 部分没有为 NAL 单元定义任何语法格式。来自 AVS 编码器的编码比特流是一种原始格式的比特流，因而它在网络传送之前需要转换为 NAL 单元。

3.7.1　AVS 视频流 NAL 单元的映射

NAL 单元定义的基本语法如图 3.25 所示。

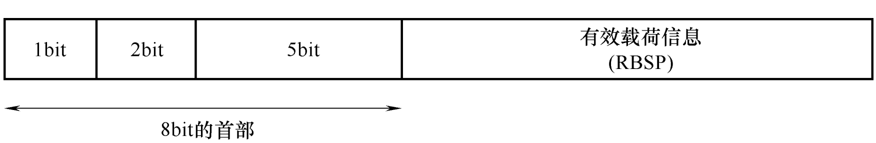


图 3.25　NAL 单元语法

具有 NAL 单元的 AVS 视频流映射程序把在 AVS 第 2 部分视频比特流中每个起始码前缀“0x000001”之间的数据映射到一个 NAL 单元（包括起始码值但不包括

码前缀）中去，然后在起始码值之前添加一个一字节的 NAL 单元。

3.7.2　NAL 单元首部描述

NAL 单元首部如图 3.26 所描述。

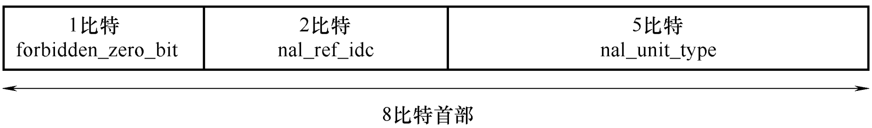


图 3.26　NAL 单元首部描述

按照 3.7.1 部分所定义的程序，视频流映射到 NAL 单元流中［H3］。位于每两个连续 NAL 首部之间的数据被认为是 RBSP 字节。用于 NAL 单元首部的语法和语义如下所述。

1. 禁用的零比特或禁用比特

其值应该总是“0”.

2. nal_ ref_ idc 或 NAL 参考 ID

NAL 参考 ID 是一个 2bit 的无符号整数。非零值表明这个 NAL 单元包含的数据是序列首部或参考帧数据；反之，零值表明该 NAL 单元包含的数据不是参考帧数据。I 帧的 Nal_ ref_ idc 值不应该是零。基于起始码类型（最大的 2 比特优先级值是“1”，最小值是“00”），Nal_ ref_ idc 指明了 NAL 单元承载的数据类型优先级。表 3.6 中示出了基于起始码类型指定在 NAL 单元首部中的 nal_ ref_ idc 值。

表 3.6　基于起始码值的 Nal_ ref_ idc（优先级）值

起始码类型	Nal_ ref_ idc 值（2bit）
序列首部	11
I 帧首部	11
I 帧的分片数据	11
P 帧的首部	10
P 帧的分片数据	10
B 帧的首部	01
B 帧的分片数据	01
视频扩展起始码	00
用户数据起始码	00
视频编辑起始码	00

3. nal_ unit_ type 或 NAL 单元类型

NAL 单元类型是一个 5bit 的无符号整数，因而允许有 32 种 NAL 单元类型。

Nal_unit_type 按照所跟随的起始码值和/或首部中包含的信息定义了 RBSP（有效载荷）数据结构的类型。表 3.7 按照起始码值示出了不同 NAL 单元类型。

表 3.7　按照起始码值和/或帧编码类型划分的 NAL 单元类型

Nal_unit_type	NAL 类型	填充原因
0	保留	
1	序列首部	起始码值是 B0
2	视频扩展	起始码值是 B5
3	用户数据	起始码值是 B2
4	视频编辑	起始码值是 B7
5	I 帧图像	起始码值是 B3
6	P 帧图像首部	起始码值是 B6，图像首部的编码模式是 01
7	B 帧图像首部	起始码值是 B6，图像首部的编码模式是 10
8	I 帧分片	起始码值是 00 ~ AF，所属图像的图像首部的编码模式是 B3
9	P 帧分片	起始码值是 00 ~ AF，所属图像的图像首部起始码值是 B6，图像首部里的编码模式是 01_2
10	B 帧分片	起始码值是 00 ~ AF，所属图像的图像首部起始码值是 B6，图像首部里的编码模式是 01_2
11 ~ 23	保留	
24 ~ 31	未定义	

4. RBSP 字节

RBSP 字节用于有效载荷数据字节的对齐，这些数据包括起始码值但不包括起始码前缀。

5. NAL 单元定界符

每个 NAL 单元之后会加上一个 4 字节的 NAL 单元定界符序列。其值是“0000 0000 0000 0000 0000 0000 0000 0001”或“0x000001”（译者认为是 0x00000001）。加入 NAL 单元定界符只是辅助确定 NAL 单元的起始位置。在解码过程中定界符就被丢弃。

只有当 AVS 第 2 部分视频比特流以这种格式完成到 NAL 单元流的映射，视频比特流才能用来在网络上传输。在接收到 NAL 单元流之后，为了解码，NAL 单元首部和 NAL 单元定界符要被丢弃并代之以值为‘0x000001’的起始码前缀，把 NAL 单元流转换回 AVS 第 2 部分视频的原始比特流。

3.8　AVS-M 简介（AVS 第 7 部分）

AVS-M 是中国音频视频编码标准（AVS）工作组制定的系列标准的第七部分，

重点服务于计算能力和功耗有限的移动系统及设备的视频编码。AVS-M 标准能够涵盖广泛的应用范围，包括移动多媒体广播、IP（网际协议）多媒体系统（IMS）、多媒体邮件、分组网络上的多媒体业务、视频电话和视频监控。

在此标准里，使用了大量的技术来实现高效的压缩，包括帧间预测、帧内预测、变换、量化、熵编码等。帧间编码利用基于块帧间预测中的运动矢量来开发相邻帧间的时间统计相关性。帧内编码利用各种空间预测模式来开发一个单帧内源信号的空间统计相关性。然后，利用一个变换在量化之前去除变换块内的空间相关，将预测残差进一步压缩。这样就产生了一个不可逆的过程，该过程一般会丢弃那些不重要的视觉信息，同时形成了源样值的紧密近似值。最后，运动矢量或帧内预测模式与量化系数进行复用并进行熵编码。

3.8.1　AVS-M 的数据结构

AVS-M 具有两种比特流格式：网络抽象层（NAL）和单元流格式或字节流格式。NAL 单元流包括一个叫作 NAL 单元的语法结构序列。字节流格式可由 NAL 单元流构造，NAL 单元流包括按照解码顺序排列的 NAL 单元序列。每个 NAL 单元以起始码作为前缀。

视频序列按照层进行编码：编码的视频序列、帧、分片、宏块和块，其中视频序列是最高层块是最低层。较高级的层包括较低级的层及其首部。

1. 编码的视频序列［A74］

编码的视频序列是比特流中最高级的结构，其比特流以 IDR（Instantaneous Decode Refresh，即时解码刷新）帧开始，后面是零或者更多的非 IDR 帧直到不包括下一个 IDR 帧或比特流的结尾。解码帧在比特流中按顺序排列，与解码顺序相同。

2. 帧［A74］

帧表示一个序列中的一幅图像。帧的码流包括一个帧首部和三个数据矩阵：Y（亮度），Cb（蓝色差）和 Cr（红色差）以及 Y，Cb，Cr 和原始的模拟分量（红、绿、蓝）之间的关系。解码器输出一系列帧。两个连续帧之间有一个时间间隔，叫作一帧的持续时间。帧解码过程包括分解（首部和矩阵信息）过程和解码过程。该标准支持 4:2:0 格式。在此格式中，Cb 和 Cr 矩阵在水平和垂直维度上都是 Y 矩阵的一半大小，如图 3.27 所示。“O”表示亮度样值 Y 的位置而“X”表示色度样值 Cb 和 Cr 的位置。

该标准规定了两类解码帧：内部解码帧（I-帧）和前向帧间解码帧（P-帧）。P-帧最多可有两个参考帧用于前向预测。运动矢量可有超越参考帧的边界。在此情况下，帧内最近的像素将被用于边界扩展。对于亮度样值，参考像素块不会超过帧边界 16 个像素；对于色度样值，在水平和垂直方向上都不会超过 8 个像素。

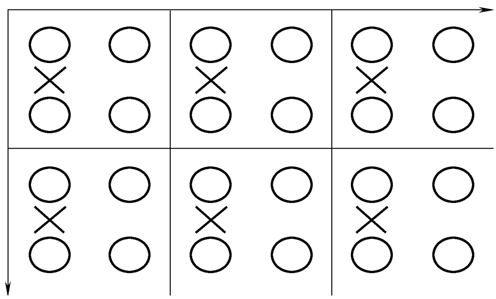


图 3. 27　4∶2∶0 格式帧内亮度和色度样值的水平和垂直位置示意图［A74］

3. 分片

分片是一个以光栅扫描顺序连续排列的宏块序列。分片内的宏块（MB，Macroblock）不能互相重叠而且分片之间也不互相重叠。分片涵盖真个帧区域，除了环路滤波过程之外，分片内的宏块解码过程不会用到帧内其他分片的数据。帧分解为分片的示例如图 3. 28 所示。

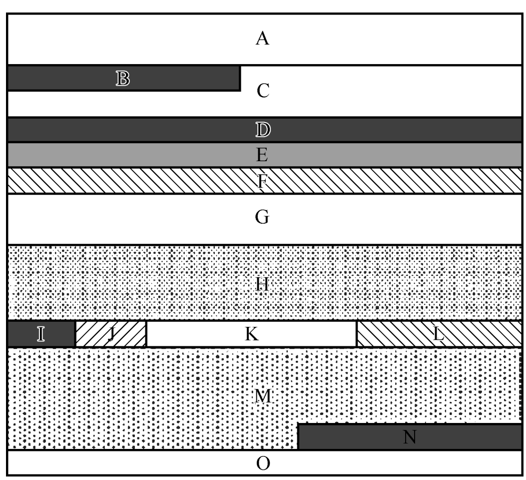


图 3. 28　分片［A74］

4. 宏块

分片细分为宏块。宏块是一个大小为 16 × 16 的像素区块。每个宏块的左上角样值不会超过帧边界。这种划分用于运动补偿。每个矩形中的数字规定了运动矢量出现的顺序以及比特流中的参考标号。将帧划分为分片使得两个分片的宏块没有重叠，对一个宏块而言分片也没有重叠。这一点必须要确保才能保证一个分片中的所有宏块可以只用相同分片中的宏块作为相邻块进行解码。

一个宏块包括 4 个 8 × 8 的亮度块（Y）和两个 8 × 8 的色度块（一个 Cb 和一个 Cr）如图 3. 29 所示。每个矩形框内的数字规定了比特流内一个宏块中每个 8 × 8 块的编码顺序。

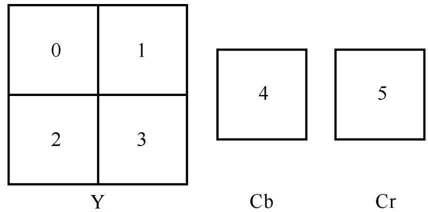


图 3.29 4:2:0 格式的宏块结构

如果宏块以子块类型进行编码，那么每个 8×8 块要分为大小从 8×8 到 4×4 的子块。图 3.30 所示为一个宏块分解为 16×4×4 的亮度块，两个色度块每个分成 4×4×4 块。方框里的数字表示编码宏块时的扫描顺序。

编码视频序列比特流中的分层数据总体结构如图 3.31 所示。

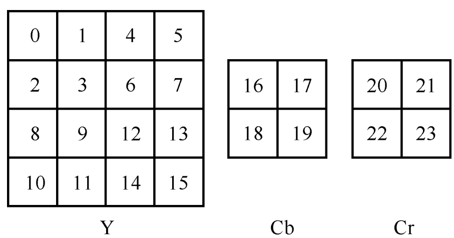


图 3.30 4:2:0 格式宏块内扫描顺序［A74］

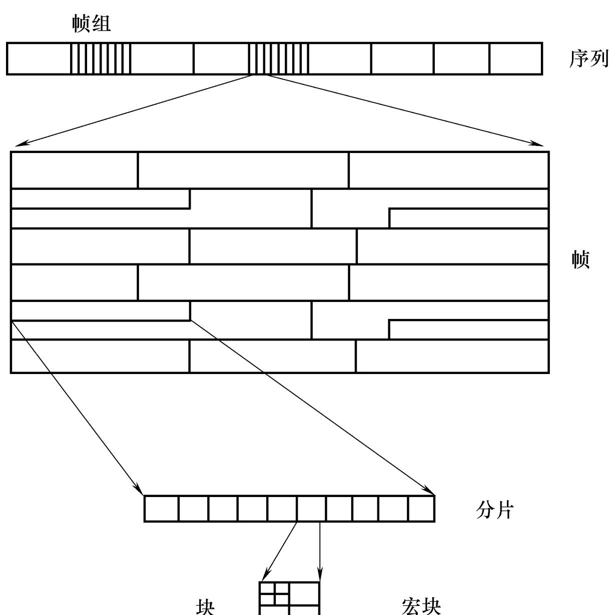


图 3.31 AVS-M 的数据结构

3.8.2　AVS-M 的体现

AVS-M 编码器和解码器［B17］的框图分别如图 3.32 和图 3.33 所示。宏块是需要预测的（帧内预测或帧间预测）。在 AVS-M 编码器中，S0 用来为当前宏块选择右预测方法，而在解码器中 S0 由当前宏块的宏块类型所控制。然后，从原始宏块中减去预测宏块得到预测残差。之后，对残差进行 ICT（整数余弦变换）再量化。量化系数与运动矢量（如果宏块是帧间预测的）一起用 2D VLC 进行熵编码，然后传输到解码端。编码器在自己这一端也保有一个局部解码器以得到与解码端获得的同样的重建帧。这样做的目的在于编码器用来预测的帧正好是解码器用来进行重建的那一帧。所以，编码器和解码器是同步工作的。如果不能保证这一点，那么量化误差就会逐渐积累。

一旦解码器接收到 AVS-M 比特流，它就对比特流进行熵解码，如果宏块是帧间预测的就要分离运动矢量，然后将逆变换、量化之后的系数加到预测（帧间或帧内预测）宏块上，再对边缘施加去块效应滤波。这样，视频帧就准备好由任何播放器进行播放了。重建的视频帧也被存储在一个帧缓存里作为未来帧帧间预测的参考帧，和同一帧里未来宏块的帧内预测的参考帧。解码器的框图如图 3.33 所示。

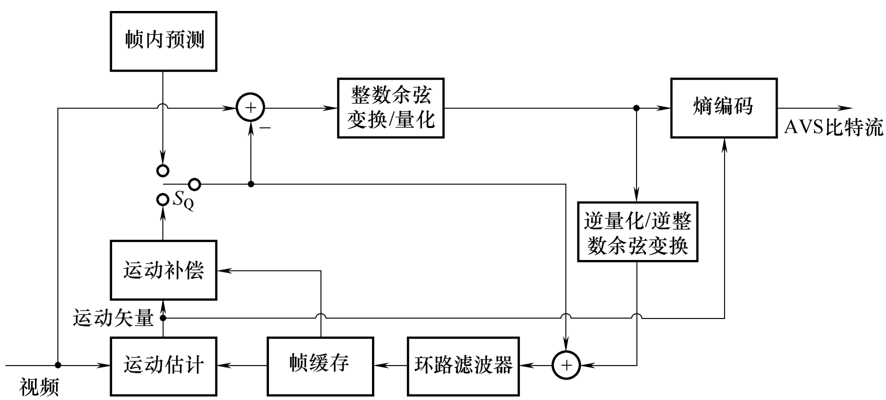


图 3.32　AVS-M 编码器框图［B17］

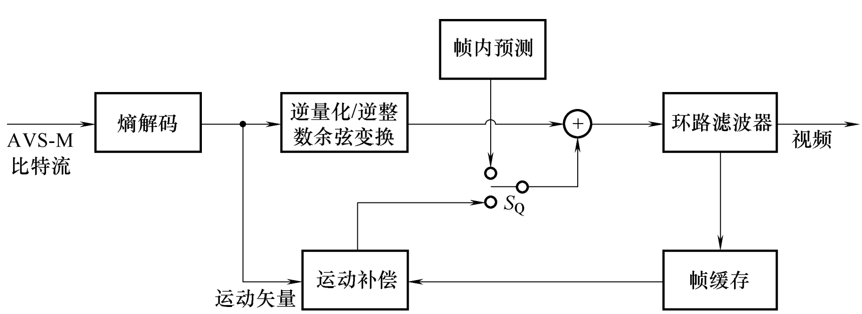


图 3.33　AVS-M 解码器的框图

3.8.3 基本档次的多个级别

这里考虑的 AVS 部分是 AVS-M，也叫作基本档次。AVS-M 用 9 个级别定义了基本档次。表 3.8 ~ 表 3.10 规定了特定级别某些语法元素的限制。

表 3.8 级别 1 的某些语法元素值的限制

参数	级别			
	1.0	1.1	1.2	1.3
每秒最大宏块数	1485	1485	6000	11880
每帧最大宏块数	99	99	396	396
最大宏块编码比特率（bit/s）	4096	4096	4096	4096
最大比特率（bit/s）	64000	128000	384000	786000
最大解码帧缓存大小（B）	114048	114048	456192	456192
最大编码帧缓存大小（bit）	17500	350000	1000000	2000000
运动矢量最大水平范围	[-2048, 2047.75]	[-2048, 2047.75]	[-2048, 2047.75]	[-2048, 2047.75]
运动矢量最大垂直范围	[-32, 31.75]	[-32, 31.75]	[-32, 31.75]	[-32, 31.75]
最大子宏块矩形大小（B）	572	572	572	572
最小压缩比	2	2	2	2

表 3.9 级别 2 基本档次的语法元素限制

参数	级别		
	2.0	2.1	2.2
每秒最大宏块数	11880	19800	20250
每帧最大宏块数	396	792	1620
最大宏块编码比特率（bit/s）	4096	4096	4096
最大比特率（bit/s）	2000000	4000000	4000000
最大解码帧缓存大小（B）	456192	912384	15552000
最大编码帧缓存大小（bit）	2000000	4000000	4000000

（续）

参　数	级　别		
	2.0	2.1	2.2
运动矢量最大水平范围	[-2048，2047.75]	[-2048，2047.75]	[-2048，2047.75]
运动矢量最大垂直范围	[-32，31.75]	[-32，31.75]	[-32，31.75]
最大子宏块矩形大小（B）	572	572	572
最小压缩比	2	2	2

表 3.10　3.X 级别的基本档次语法元素限制

参　数	级　别	
	3.0	3.1
每秒最大宏块数	36000	40500
每帧最大宏块数	1620	1620
最大宏块编码比特率（bit/s）	4096	4096
最大比特率（bit/s）	6000000	8000000
最大解码帧缓存大小（B）	1555200	1555200
最大编码帧缓存大小（bit）	6000000	8000000
运动矢量最大水平范围	[-2048，2047.75]	[-2048，2047.75]
运动矢量最大垂直范围	[-256，255.75]	[-256，255.75]
最大子宏块矩形大小（B）	572	572
最小压缩比	2	2

3.9　块模式预测模式

3.9.1　帧内预测

帧内预测结果从左侧和上方的块的相邻像素得出。有 9 种帧内 4×4 模式如图 3.34所示。帧内预测单元的大小是 4×4，因为 AVS-M 使用 4×4 的整数余弦变换（ICT）。某些特定的方法与 4×4 帧内预测一起工作，诸如直接帧内预测（DIP，Direct Intra Prediction），预测前的填充（PBP，Padding Before Prediction）和简化的色度帧内预测（SCI，Simplified Chrominance Intra-prediction）[A59]。也会用到来自相邻块的最可能模式预测 [A59]。表 3.11 示出了当前块基于左侧（L）和上方

（U）块的最可能模式［A27］。一旦没有左侧块或上方块或该块是帧间编码的，L 或R 就被指定为模式“－1”。一个块级标志指示特定的块是否使用“最可能模式”进行编码，一个宏块级标志指示使用了 DIP［A75，A76］。如果一个宏块标记为 DIP 模式，这意味着在该宏块里的 16 个 4×4 子块每一个都采取最可能模式作为帧内预测模式，即使每个 4×4 子块的帧内预测模式可能不同，这样在比特流中就必传输更多的模式信息，从而节约了大量的比特。最可能模式的发生概率是非常高的［A24］，这样就节约了整个过程的许多比特。PBP 用于亮度和色度分量，在此过程中参考像素“Ds”和“Hs”分别从“D”和“H”填充得到，这样就跳过了右上和左下方参考像素可用性的条件测试。SCI 意味着对于色度分量只使用直流、垂直和水平模式［A77］。

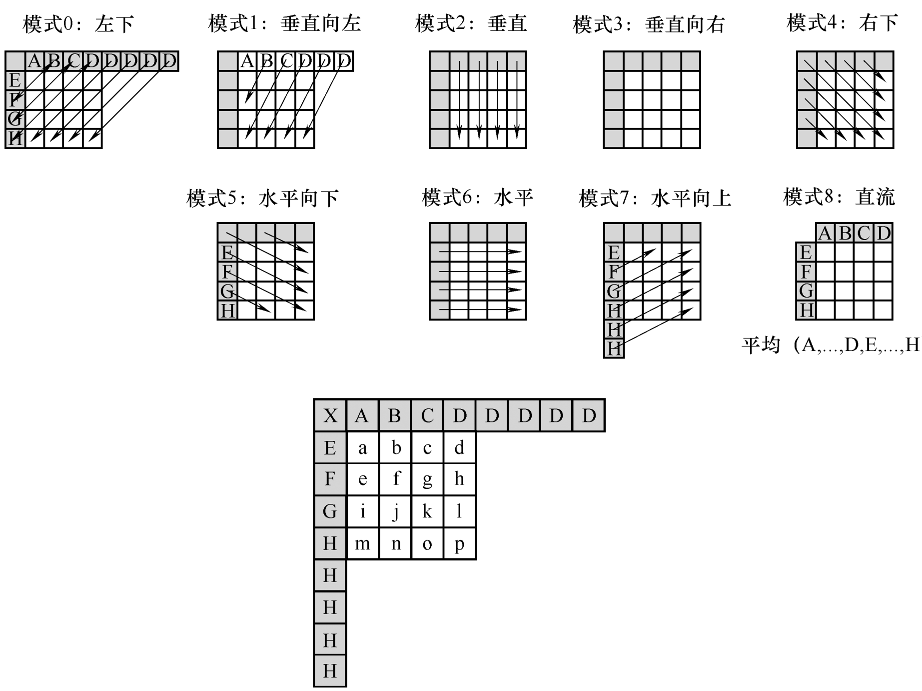


图 3.34　帧内预测模式［A27］

自适应帧内预测使得 4×4 帧内预测能和 8×8 帧内预测一起使用，只要在 AVS-2 中宏块首部里使用一个指示值［A78］。此外，如果当前块及其相邻块使用了不同块大小的帧内预测，那么在最可能模式之前需要在 4×4 帧内预测模式和 8×8帧内预测模式直接建立映射，

如果当前块是一个色度块，那么预测模式 Intra_Chroma_Pred_Mode 等于 intra_Chroma_pred_mode。每个宏块 4×4 的色度块使用相同的预测模式。4×4 色度帧内预测模式如图 3.35 所示。

表 3.11　最可能模式表［A27］

L	U									
	–1	0	1	2	3	4	5	6	7	8
–1	8	8	8	8	8	8	8	8	8	8
0	8	0	0	2	0	0	0	2	0	2
1	8	2	1	2	2	2	2	2	2	2
2	8	2	2	2	2	2	2	2	2	2
3	8	2	1	2	3	4	5	2	7	2
4	8	4	4	2	4	4	4	6	4	4
5	8	5	5	2	5	5	5	6	5	5
6	8	6	6	6	6	6	6	6	6	6
7	8	7	7	2	7	7	7	6	7	7
8	8	0	1	2	2	4	5	6	7	8

（L 表示左侧块的帧内模式，U 表示上方块的帧内模式）

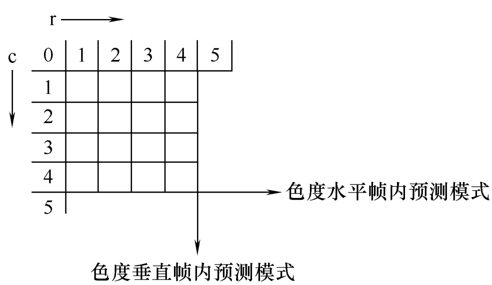


图 3.35　AVS-M 中的 intra_Chroma_pred_mode 模式［A27］

3.9.2　帧间预测

为了去除视频序列中的时间冗余，帧间预测从以前解码的帧/场进行预测。许多方法共同为 AVS-video 帧间预测的编码效率做出贡献。有两类帧间预测：P-预测和 Bi-预测。AVS-M 只支持 P-预测，它只使用过去的解码帧作为参考帧。

帧间预测结果来自于解码帧。在 AVS-M 中共支持 7 种类型的大小，16×16、16×8、8×16、8×8、8×4、4×8 和 4×4，如图 3.36 所示［A18］。请注意，AVS 第 2 部分并不支持子宏块分割。帧间预测运动矢量的精度最高为 1/4 像素，这是因为 AVS-video 的运动补偿预测允许运动矢量精度最高为 1/4 像素，相应的分数运动矢量的参考像素值由亚像素内插获得，如图 3.37 所示。AVS-video 默认的亚像素内插叫作两步四抽头（TSFT，two steps four taps）内插［A8］，且将三种滤波器分

别作用于不同位置的亚像素上。两步四抽头内插在第一步使用系数为（-1，5，5，-1）的滤波器来获得半像素的参考像素值。第二步，利用系数为（1，7，7，1）的滤波器水平地或垂直地求取1/4像素的参考像素值。第二步的例外是对于e、g、p、r的1/4像素参考像素值，使用了对角双线性滤波器。

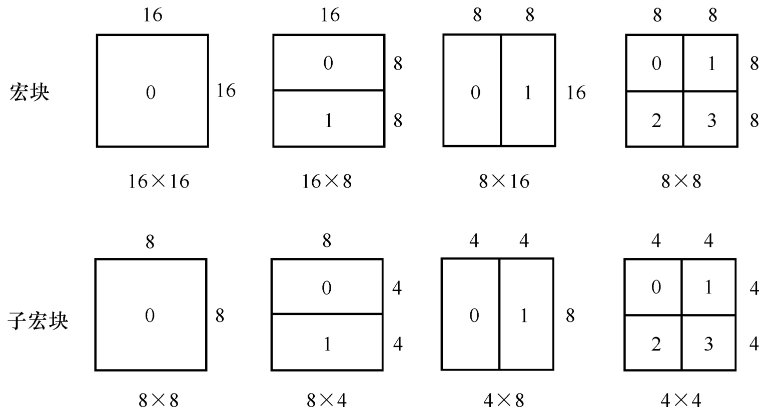


图3.36 帧间预测块大小

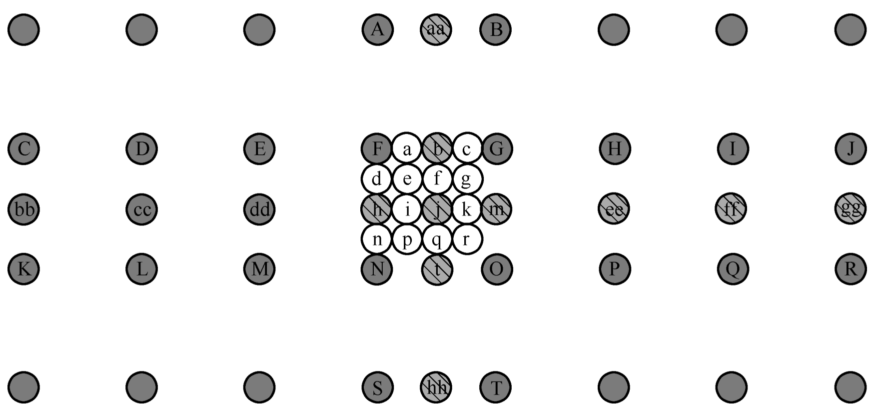


图3.37 整像素、半像素、1/4参考像素的位置

小尺寸块的性能好于大尺寸块，因为分辨率较低。4×4是变换和帧内预测的单元［A6］［A14］，也是AVS第7部分运动补偿的最小块尺寸。

3.9.3 跳跃模式预测

当一个宏块被确定用跳跃模式进行编码时，除了以跳跃模式编码的标记外，不发送该宏块的其他信息。那么，当解码器看到该宏块用跳跃模式进行编码时，解码器就会使用缓存器中标记为零的默认参考，以及基于相邻宏块运动矢量（左侧宏块、上方宏块）计算得到的运动矢量［A74］。

3.9.4 率失真优化

对于 I 帧，对每个内部块模式计算一个率失真（RD，Rate Distortion）代价，由式（3.2）从 9 个可用的内部模式中来选择最优模式。

$$\text{RD-Cost}(\text{模式}) = \text{D}(\text{模式}) + \lambda \times \text{R}(\text{模式}) \tag{3.2}$$

式中，λ 是拉格朗日乘子，基于率-代价曲线优化得出；RD-Cost（模式）是一个块特定模式的率失真代价；D（模式）表示如果该块用该模式编码的失真；R（模式）是该块用那个特定模式编码产生的比特率。所以，为了确定一个块的块模式所有 9 种模式代价都要计算。为了计算每个代价，编码器需要对一个块用所有的模式进行变换、量化和熵编码来计算 R（模式），因为 R（模式）是用来编码一个块的比特量。同样，编码器在自己这一端也必须执行熵解码、逆量化和逆变换来重建图像，以计算 D（模式）因为 D（模式）是原始图像和重建图像直接的差值。宏块的全部 16 个块最好的 RD-Cost 计算完毕，如果所有块都以 MPM（most probable mode，最可能模式）编码，编码器就会计算宏块的 RD-Cost，如果它碰巧小于全部 16 个块最好的 RD-Cost 值，就用 DIP（直接帧内预测，direct intra prediction）来编码该宏块。

对于 P 帧，编码器计算所有帧间模式的代价，为帧间预测计算出最优模式。基于 R-D 优化选定最优的帧间模式，然后也为跳跃宏块计算出 R-D 代价。具有最低代价的编码模式被选定，宏块就用那个模式来编码。

3.10 变换、量化和熵编码

3.10.1 变换

运动补偿的残差用 4×4 ICT（整数余弦变换）进行变换，AVS 中使用的 4×4 变换是［A74］：

$$T_4 = \begin{bmatrix} 2 & 3 & 2 & 1 \\ 2 & 1 & -2 & -3 \\ 2 & -1 & -2 & 3 \\ 2 & -3 & 2 & -1 \end{bmatrix}$$

ICT 是可分离的且具有整数精度。设计它是为了最小化解码器的实现复杂度。

3.10.2 量化

一个自适应均匀量化器用于在 4×4 的系数矩阵上执行量化过程［B17，A6，A27］。量化器的步长是可变的以提供率控制功能。在固定比特率工作条件下，该机制可用于防止缓存器溢出。传输的步长量化参数（QP）直接用于亮度系数，而

对于色度系数它的范围上界需要修改。对于整帧或分片，量化参数可选为固定值。如果量化参数不固定，它可在每个宏块上进行不同的更新。量化参数从 0 到 63 不等，步长为 1。为了提供低复杂度的解码器实现方案，人们修改均匀量化过程来和变换过程协同工作。量化系数以 Z 形方式进行扫描，如图 3.38 所示［A6］。

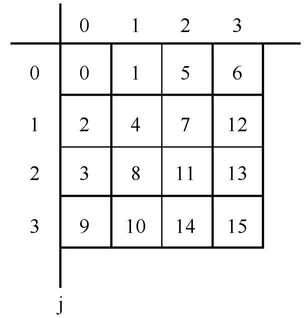


图 3.38　用于量化变换系数的 Z 形扫描模式

3.10.3　熵编码

AVS-M 使用指数 Golomb 码［A27］进行熵编码，见表 3.12，对语法元素诸如量化系数，宏块编码类型，和运动矢量进行编码。有 18 个表用于编码量化系数［A59］。编码器用游程和当前系数的绝对值来选择表 3.12。AVS-M 使用了自适应变长编码（VLC）技术［B17］。

重建帧是预测帧和当前重建误差帧之和。在图 3.32 和图 3.33，编码器将逆量化和逆变换系数加到预测帧（通过 S0 开关选择的帧内预测或帧间预测）上以得到重建帧。AVS-M 在重建帧上使用去块效应滤波器，然后将它们存储在缓存器中作为未来的参考。

表 3.12　第 k 阶 Golomb 编码

指　　数	码　结　构	码数字范围
$k=0$	1	0
	$01x_0$	1 ~ 2
	$001x_1x_0$	3 ~ 6
	$0001x_2x_1x_0$	7 ~ 14
	…	…
$k=1$	$1x_0$	0 ~ 1
	$01x_1x_0$	2 ~ 5
	$001x_2x_1x_0$	6 ~ 13
	$0001x_3x_2x_1x_0$	14 ~ 29
	…	…

3.10.4　简化的去块效应滤波器

去块效应过程首先沿着垂直边沿然后沿着水平边沿直接作用于重建的参考帧上。因为不同的图像区域和不同的比特率需要不同的平滑类型，所以 AVS-M 依据块的活动性和量化参数对去块效应滤波器进行了相应的调整。除了帧边界的边沿和 disable_loop_filter_slice_flag 等于 1 的分片边沿，所有宏块的 4×4 边沿都要被滤波

处理。环路滤波器把一个宏块看作一个单元。每个宏块处理如下：

亮度和色度的滤波是分开的，如图 3.39 所示。首先从左到右进行垂直滤波，然后从上到下进行水平滤波。尚未被环路滤波过程修改的样值用作当前宏块滤波过程的输入。水平边沿滤波过程的输入是当前宏块在垂直边沿滤波过程［A74］中已经修改过的样值。

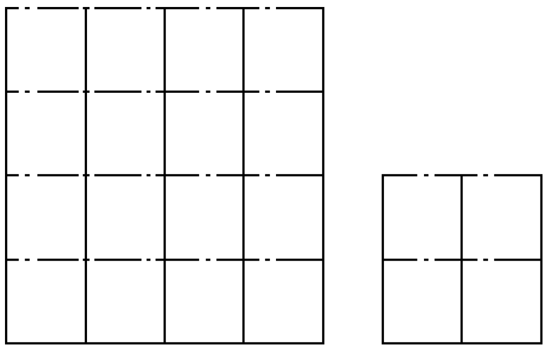


图 3.39　待滤波宏块的块边沿［A74］

AVS 第 7 部分利用了一个简化的去块效应滤波器，边界强度在宏块级别进行确定。滤波作用于帧或分片边界之外的亮度和色度块的边界。帧内预测宏块通常比帧间预测宏块有更多更大的残差，这导致了在某些 QP 上非常明显的块效应。因此，对于帧内预测宏块使用强效的滤波器，对于帧间预测宏块使用弱效的滤波器。宏块类型为 P_Skip 的时候，没有编码残差。QP 不太大的时候，由量化引起的失真相对小一些，后面就不需要滤波。

基于宏块类型和 QP，标准使用了三种滤波模式。

对于帧内预测宏块，对宏块中的每个块边界以及宏块的左上边界使用内部滤波模式。对于不是 P_Skip 类型或 QP 大于某个阈值的帧间预测宏块，使用帧间滤波器模式。对于 P_Skip 且 QP 小于某个阈值的帧间预测宏块，绕过滤波过程。

对于既是帧内又是帧间模式的滤波边沿，进行样值级别的滤波判决。如果下列三个条件很好地成立，那么就进行滤波过程；否则绕过滤波过程。

$$|p_0 - q_0| < \alpha(\text{IndexA}) \tag{3.3}$$

$$|p_1 - p_0| < \beta(\text{IndexB}) \tag{3.4}$$

$$|q_1 - q_0| < \beta(\text{IndexB}) \tag{3.5}$$

式中，α 和 β 可由 IndexA，IndexB 计算得出。α 是 IndexA 的一个函数，依此类推。p_1，p_0，q_1 和 q_0 是跨越每个样值级别边界的样值（见图 3.40）。

对于帧内和帧间滤波模式，滤波过程是不同的。对于帧内滤波模式，用下列方程计算每个待滤波块的 Δ_0 和 Δ_1。

$$\Delta_0 = \text{Clip}(-C_1, C_1, ((q_0 - p_0) \times 4 + (p_1 - q_1) + 4) >> 3) \tag{3.6}$$

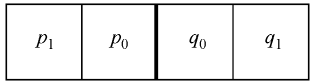


图 3.40　AVS 去块效应滤波器样值级别的边界［A24］

$$\Delta_1 = \Delta_0 >> 1 \tag{3.7}$$

函数 Clip 是一个定义为 Clip(a, b, c) = min[max(a, c), b] 的运算。C_I 是 I 分片 Clip 函数的上限，通过查表得到。

下列公式用来得到 p'_0，p'_1，q'_0 和 q'_1 的值。

$$p'_0 = p_0 + \Delta_0 \tag{3.8}$$

$$p'_1 = p_1 + \Delta_1 \tag{3.9}$$

$$q'_0 = q_0 - \Delta_0 \tag{3.10}$$

$$q'_1 = q_1 - \Delta_1 \tag{3.11}$$

最后，p_1，p_0，q_1 和 q_0 的值由范围是 0- 255 的 p'_0、p'_1、q'_0 和 q'_1 来代替。

对于帧间滤波模式，首先用 CP 代替 CI 计算 Δ_0 和 Δ_1。CP 是 P 分片 Clip 函数的上限值，可从 CI 推导得到。CP = CI >> 1 + CP_offset，其中 CP_offset 是比特率中传输的偏移值。p'_0，p'_1，q'_0 和 q'_1 的值可由下面的程序计算得到。

```
if(abs(p0 - p1) < (β >> 1)){
        p'0 = p0 + Δ0;
        p'1 = p1 + Δ1;}
else {
        p'0 = p0 + Δ1;
        p'1 = p1;}
if(abs(q0 - q1) < (β >> 1)){
        q'0 = q0 - Δ0;
        q'1 = q1 - Δ1;}
else {
        q'0 = q0 - Δ1;
        q'1 = q1;}
```

3.11　AVS 第 1 部分：系统

这部分内容来自中国的 AVS 标准《GB/T 20090.1 信息技术　先进音视频编码　第 1 部分：系统》。

AVS 第 1 部分［A1］将单/多通道音频视频流或其他基本数据流转换成单/多通道复用流以备存储和传输。

AVS 的系统编码包括两个部分：运输流和程序流，对于不同的应用要进行优

化。AVS 定义了编码语法，这对于同步解码和显示音频视频信息是必要的。该语法确保解码器的数据缓存器中不会出现溢出和下溢现象。该语法至少包括下列两类信息之一：

a）描述被编码的音频/视频数据和流比特率的相对时间戳信息；

b）描述被编码音频/视频数据的时间戳和流传输的时间戳信息。

同时，运输流和程序流也是包的复用流。

单个音频和视频基本流的复用过程如图 3.41 所示。视频和音频可以按照 AVS 第 2 部分、第 7 部分、第 3 部分和第 10 部分编码标准进行编码。然后，压缩数据被打包成为一个 PES。

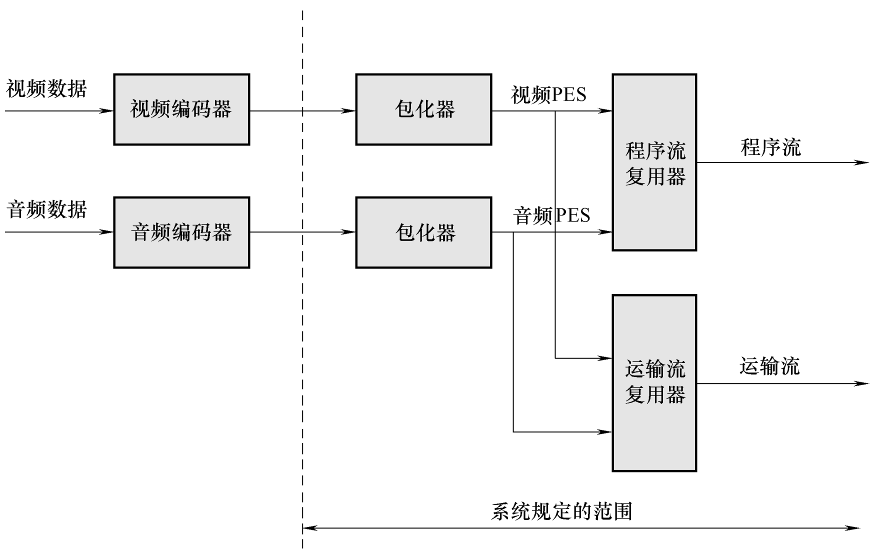


图 3.41　复用过程图

PES 的语法结构如图 3.42 所示。

PES 是包化基本流（packetized elementary stream）的缩写。它的语法和语义将被用于封装音频和视频基本流信息。PES 语法用于传递显示时间戳（PTS，Presentation Time-Stamp）和解码时间戳（DTS，Decoding Time-Stamp）信息，同步解码音频和视频信息需要这些信息。（见图 3.43）。

3.11.1　程序流

程序流是包括一个或更多 PES 包的单独的流，这些 PES 包具有共同的时间戳，为相对无错的环境而设计，适应于互动多媒体应用等。程序流的包长度是可变的。

程序流的速率可以固定也可以变化。在任何情况下，程序流包含的基本流的速

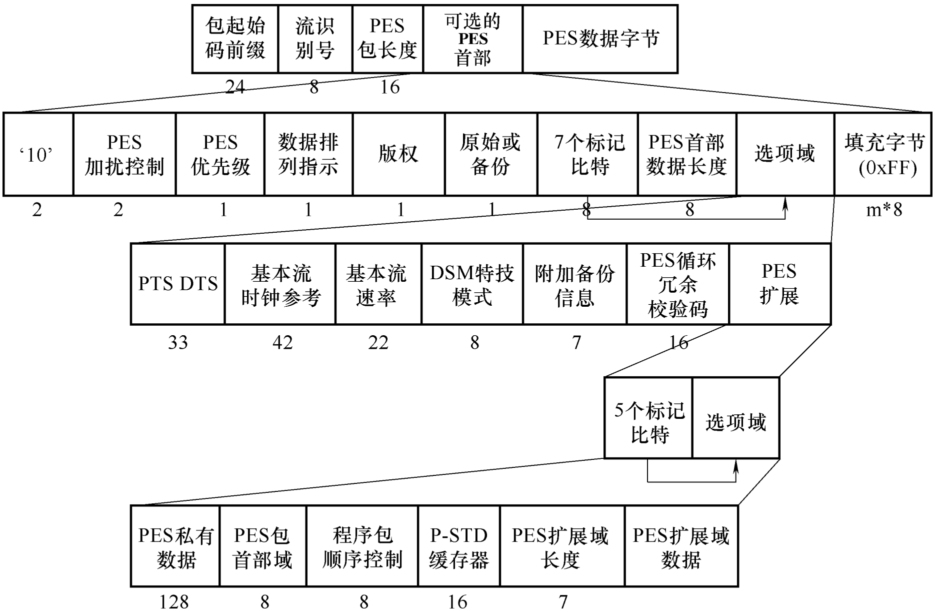


图 3.42 包化基本流的语法

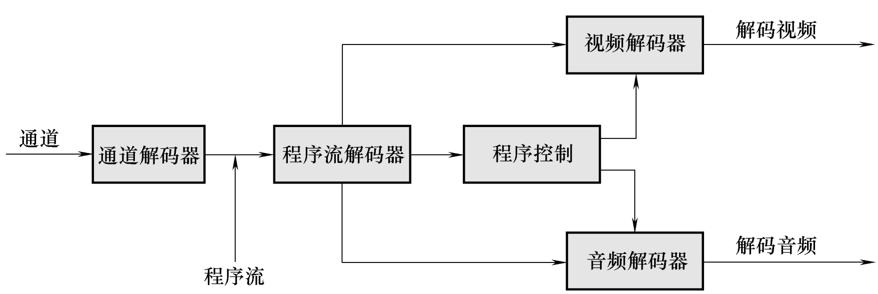


图 3.43 程序流解码器

率是恒定或可变的。程序流的速率由系统时钟参考速率（SCR，System Clock Reference Rate）域和复用速率域的值和位置来定义。

图 3.43 所示为一个 AVS 标准音频/视频程序流解码系统的原型，但是其结构不唯一。图 3.44 所示的程序流解码器原型包括系统，视频和音频，与 AVS 第 1 部分、第 2 部分和第 3 部分相一致。

程序流原型解码器接受程序流作为输入，依靠程序流解码器从流中抽取信息。然后，程序流解码器对复用流进行解复用，输出变为音频解码器和视频解码器输入的基本流。音频/视频解码器的输出就是解码后的音频和视频信号。

程序流可分为两层：系统层和压缩层。程序流解码器的输入流包括一个含有压缩层的系统层。对于音频解码器和视频解码器，输入流只包括压缩层。

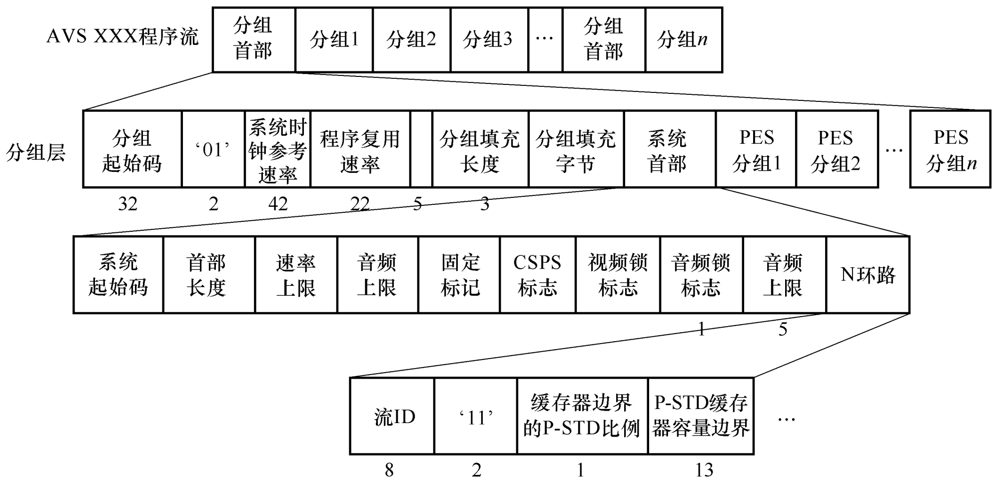


图 3.44　程序流的结构

程序流编码器可以把一个含有一个或更多基本流的程序组合成一个单独的流。在此情况下，基本流数据和能使这些流同步播放的信息复用在一起。

一个程序流包含一个或更多的复用在一起且属于相同程序的基本流。PES 包携带基本流的数据。PES 分组群的安排是，分组首部在先，包紧随其后。

PES 分组首部开始于一个 32 比特的码，指明了分组数据属于哪一个流。

在程序流中，PES 分组组合为一个群。PES 分组群开始于分组首部，接着是一些 PES 分组或者没有。分组首部存储时间和比特率信息从 32 比特的 start_code 开始。程序流的结构如图 3.44 所示。

程序流图（PSM，Program Stream Map）描述了基本流及其相互关系，如图 3.45 所示。

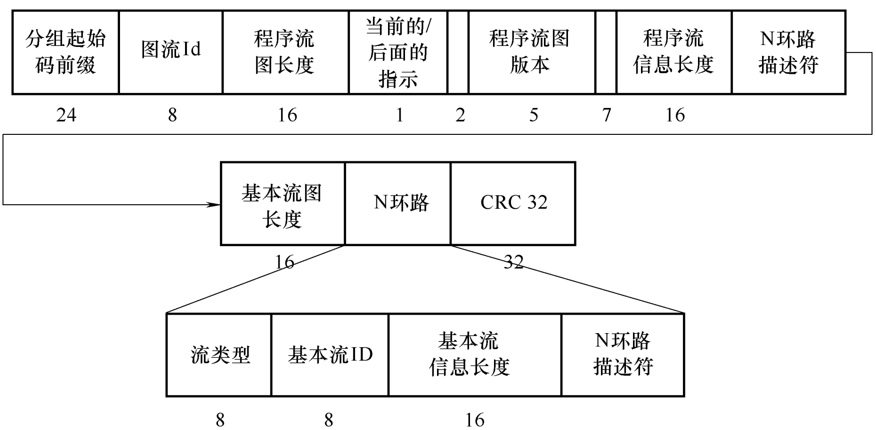


图 3.45　程序流图

3.11.2 运输流

运输流由一个或多个单独的程序组成，形成一个单一的流。这些程序可以包含一个或多个不同的时钟参考，在相同的程序中包含若干基本流的 PES 分组共享一个共同的时钟参考。设计运输流是为了抑制存储介质或传输环境中可能的有害误差。运输流的分组长度是 188B。

运输流的速率可能是恒定的或可变的。不管怎样，程序流包含的基本流也是恒定的或可变的。程序流的速率由 transport_rate 域定义，它嵌入在每个流中。

某些操作能用最小的开销来执行，这是运输流的设计原则。这些操作包括：

（a）从运输流中的程序获取编码数据，解码并进行显示；

（b）从一个程序中抽取运输流分组，生成一个只包含该程序的新流作为输出；

（c）从一个或多个运输流中抽取属于一个或多个程序的运输流分组，形成带有这些抽取分组的新运输流；

（d）从运输流这抽取程序的内容，生成带有抽取程序的程序流；

（e）将程序流转换为运输流，并在一个有损和有噪的环境中进行传输。然后重建一个有效的程序流，在某些情况下正好就是原始的程序流。

图 3.46 描述了一个以运输流为输入的解复用和解码系统原型。图 3.47 说明了操作 a，在其中一共运输流被直接解复用和解码。运输流的结构可被分为两层：系统层和压缩层。

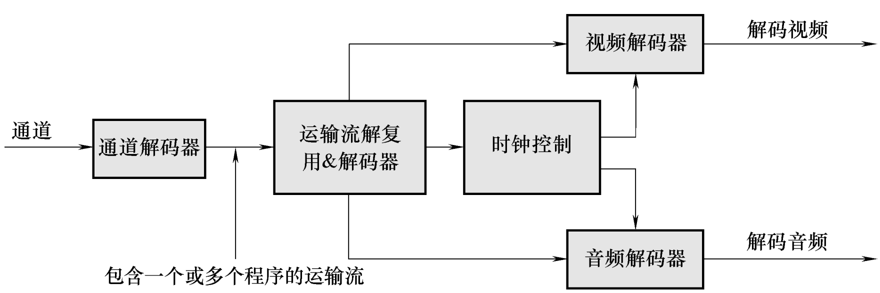


图 3.46 运输流解复用与解码原型

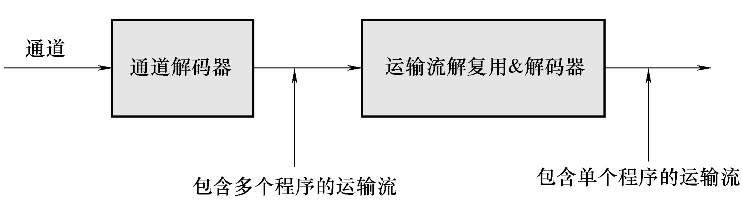


图 3.47 传输复用原型

图 3.46 说明了一个运输流解码器原型，它包括 AVS 标准提供的视频和音频解

码器。这不意味着 AVS 标准发布的原型对于运输流解码器的设计要求严格的限制，而且解码器也不是唯一的。事实上，非音频/视频数据也能被使用。

图 3.47 示出了另一种情况，其中包含一个或多个程序的运输流被变换为只包含单一程序的运输流。在此情况下，可能需要重复用操作来矫正 transport_rate 的值，该值用于补偿比特流中传输速率的变化。

AVS 运输流编码允许一个或多个程序封装在一个流中。每个基本流和用来同步数据并显示每个程序基本流的信息复用在一起。运输流如图 3.48 所示。

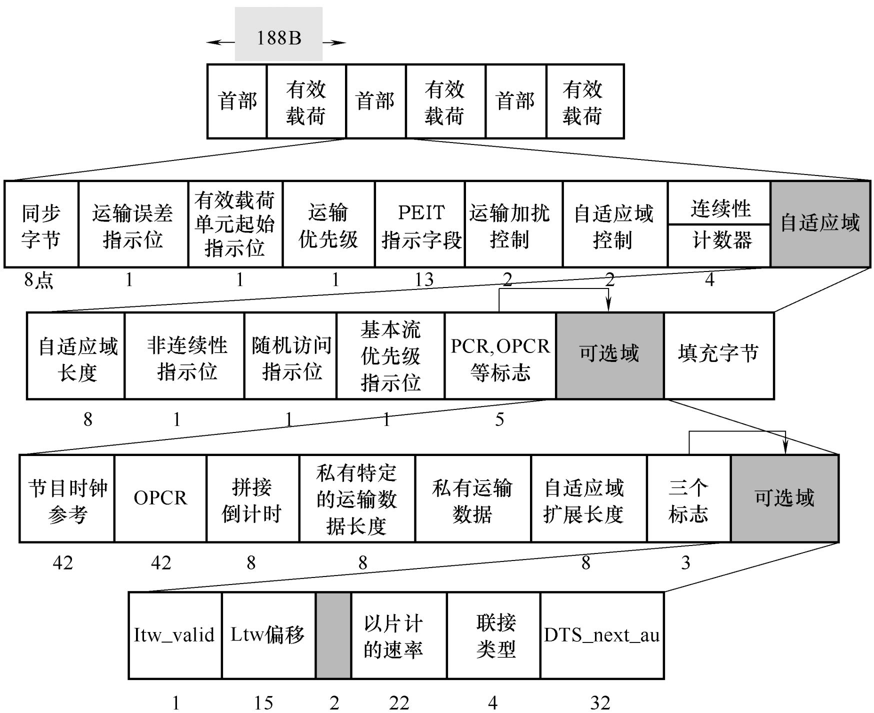


图 3.48　运输流的结构

一个运输流包括一个或多个程序。视频和音频基本流包括一个或多个访问单元。基本流数据由 PES 包携带。一个 PES 包由首部和数据组成。

PES 包插入到运输流包之内，每个 PES 包首部的第一个字节位于运输流包第一个可用的有效载荷的位置。PES 包首部以 32bit 的起始码开头，该起始码也指明该包所属的流或流类型。

运输流包可以为空，空包用来填充运输流，在重复用过程中可被插入或者删除。

运输流图描述了基本流及其相关关系，如图 3.49 所示。

设计程序流和运输流是为了满足不同的应用。程序流和运输流的定义并不严格遵守分层模型；它们彼此之间可以互相交换，但不是对方的一个子集或超集。从一

个运输流中可以抽取一个程序，一个有效的程序流可由此创建。这两种类型流之间的转换可以按照 PES 包的公共交换格式来完成，但不是程序流所需的所有域包含在运输流之中；某些域需要被导出。

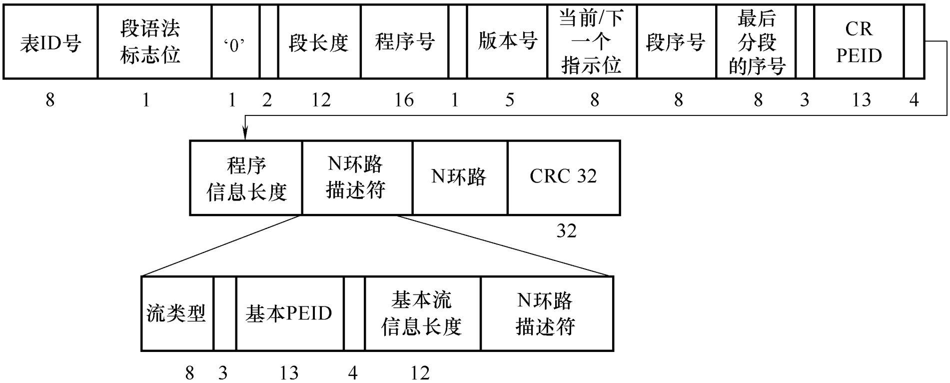


图 3.49 运输流图

3.12 IEEE AVS

P1857/D1 2012 年 7 月试用草案代表被 IEEE-SA 标准委员会通过的先进音视频编码标准。这是一个 IEEE 标准提案的获准草案，提案于 2013 年 4 月由 IEEE-SA 委员会批准通过，于 2013 年 6 月对公众发布。

下面内容摘自这个标准的试用草案。要访问整个标准，可发邮件 Stds. ipr@ieee. org，网址为 http://standards. ieee. org/index. html。

摘要：多媒体数据已经成为互联网中数据的主要部分。在包括有限复杂度和带宽的限制之下，IEEE AVS 标准［A82］规定了视频和音频压缩、解压、封装工具、以及高效地支持互联网上多媒体数据传输与存储的机制。

标准的这部分定义了一套高效视频编码的工具，包括多向帧内预测、变块大小帧间预测和基于上下文自适应二进制算术编码，等等。而且，它也规定了解码程序。

目标应用和业务包括但不限于互联网电视、用户生成的多媒体内容、基于 IP 的视频会议、基于 IP 的监控，以及具有其他视频/音频功能的业务和应用诸如数字电视广播、数字存储媒体和通信。

［IEEE AVS］为多媒体数据的压缩、解压和封装提供了完整的高质量而有效的编码工具集，以节约互联网的传输带宽或者存储空间。

［IEEE AVS］包括如下几个部分：

— 第一部分：系统

— 第二部分：视频

— 第三部分：音频

3.12.1　应用

设计这个标准是为了囊括广泛的应用领域，包括但不限于：

CATV（Cable TV on optical networks，copper，etc.）　光、铜线等网络上的有线电视

DBS（Direct broadcast satellite video services）　卫星视频直播业务

DSL（Digital subscriber line video services）　数字用户线路视频业务

DTTB（Digital terrestrial television broadcasting）　陆地数字电视广播

IPTV（Internet Protocol Television）　IP 电视

ISM（Interactive storage media（optical disks，etc.）　互动存储介质（光盘等）

MMM（Multimedia mailing）　多媒体邮件

MSPN（Multimedia services on packet networks）　分组网络多媒体业务

RTC（Real- time conversational services（videoconferencing，videophone，etc.）实时会话业务（视频会议、视频电话等）

RVS（Remote video surveillance）　远程视频监控

3.12.2　档次和级别

这个标准能提供很宽范围的比特率、分辨率和质量。考虑到内插能力，通过“档次”和“级别”约定了数量有限的语法子集。

“档次”就是标准的语法元素、语义和算法特征的一个子集。

“级别”是对语法元素和值所指定的限制集合，这些限制可被某一档次的语法元素所接受。

3.12.3　设计特征概述

在此标准中，若干技术可以被用于实现高效的压缩，包括帧内预测、帧间预测、变换、量化、熵编码等。帧间编码对于基于块的帧间预测使用运动矢量来开发不同帧间的时间统计冗余。帧内编码使用不同的空间预测模式来开发单帧源信号内的空间统计冗余。预测残差通过变换来进一步压缩以去除变换块内的空间相关性，之后进行量化生成了一个不可逆过程，该过程通常会丢弃不太重要的视觉信息而形成源信号样值的紧密近似。最后，运动矢量或帧内预测模式与量化系数复用在一起进行熵编码。

3.13　总结

本章详细陈述了 AVS China（编/解码器）。它非常类似于 H.264/AVC（第 4 章），除了某些功能被最小化从而导致了复杂度降低和可忽略的视觉质量损失。

3.14 专题研究

P.3.1 见 Sahana Devaraju 的专题报告“A study on AVS-M video standard”（UTA 网址→EE5359→Spring 2009）。AVS-M 是第 7 部分（见表 3.1），服务于处理能力和功耗有限的移动系统和设备。它使用视频测试序列（CIF 和 QCIF），将 SNR 的分贝值与每帧比特数的对应关系列成了表格。请读者用不同的测试序列并得到类似的结果；绘制 SSIM 与比特/帧的关系曲线。

P.3.2 见 S. Sridhar's 的论文“Multiplexing/De-multiplexing AVS-China video with AAC audio bit streams achieving lip sync”（UTA 网址→EE5359→List of recent theses/projects）。Sridhar 能够实现这一点，视频和音频重新播放之间的滞后时间（同步时延）至多为 10ms（表 6.2）。超过最大滞后时间 140ms，可以感受到视觉时延。请读者使用不同的视频和音频测试序列进行这项研究。

P.3.3 Pragnesh Ramolia 利用机器学习算法 C4.5 实现了一种低复杂度的 AVS-M。基于几个测试序列（QCIF），与原始的 AVS-M 编码器相比，他能够实现降低 75%~80% 的编码时间，而 PSNR 的降低可以容忍。请读者将这篇论文推广至 CIF 测试序列，并且在 AVS 第 2 部分 SD/HD 视频中也加以推广；参见结论和未来的工作。

第 4 章　H. 264/MPEG-4 先进视频编码

摘要：类似于 AVS China（第 3 章），描述了 H. 264/AVC 的档次、级别和功能。从视频压缩的视角说明了各种块的作用。由此可见 AVS China 和 H. 264/AVC 之间的相似性和差别。

关键词：H. 264/MPEG-4 第 10 部分　AVC　ITU-T　H. 264 编码器　H. 264 解码器　JM 软件　预测　变换　环路去块效应滤波器　熵编码

4.1　概述

2003 年出现的 H. 264/MPEG-4 第 10 部分先进视频编码（AVC：Advanced Video Coding）成为最新且最有效的视频编码标准之一［H51］。H. 264 标准由联合视频团队（JVT：Joint Video Team）制定，该团队包括 ITU-T（International Telecommunication Union -Telecommunication Standardization sector，国际电信联盟-电信标准部门）的 VCEG（Video Coding Experts Group，视频编码专家组），和 ISO/IEC 的 MPEG（Moving Picture Experts Group，运动图像专家组）组成［H51］。

由于 H. 264 便于一种网络友好的视频表示，它能支持各种互动的（视频电话）和非互动的应用（广播、流媒体、存储、视频点播）［H29］。它再次使用了以前编码标准的思想，诸如 MPEG-1、MPEG-2、MPEG-4 第 2 部分、H. 261、H. 262 和 H. 263［H23，B3］，并增加了许多其他的编码工具和方法，从而使得它具有优越的质量和压缩效率。

像任何其他以前的基于运动的编解码系统一样，它使用了下列基本的视频压缩原则［S10］：

- 用于降低空间相关的变换；
- 控制比特率的量化；
- 降低时间相关的运动补偿预测；
- 降低统计相关的熵编码。

H. 264 改善的编码效率可以归因于附加的编码工具和新的特征。H. 264 首次使用的部分新的和改进的方法罗列如下［H29］：

- 自适应帧内预测；
- 整数精度的小尺寸块变换；
- 多参考帧和广义的 B 帧；
- 可变的块大小；

- 1/4 像素精度的运动补偿；
- 内容自适应环路去块效应滤波器；
- 引入 CABAC（上下文自适应二进制算术编码）和 CAVLC（上下文自适应变长编码）改善的熵编码。

与以往的编码标准相比，编码效率的提高以及压缩比的提高导致了 H. 264 编码器和解码器算法较大的复杂度。为了便于在网络上传输信息而进行抗误码设计，H. 264 支持一下技术［H29］：

- 灵活的宏块排序；
- 切换分片；
- 任意的分片顺序；
- 冗余分片；
- 数据分割；
- 参数设置。

4. 2　H. 264 的档次和级别

H. 264/AVC 标准由很多编码工具构成。该标准也处理大范围的比特率、分辨率、质量、应用和业务。对于任何在给定时间点上的特定应用，并非需要所有的工具和所有的比特率。H. 264 所有的各种各样的工具是按照档次进行分组的。

4. 2. 1　H. 264 的档次

档次定义为一个编码工具的子集。它们有助于在复杂度受限时使互操作性最大化［S10，H13］。而不同的级别则定义了不同的参数例如解码图像的大小、比特率等。

H. 264 所定义的档次如下所列［H18］：

1）基本档次；

2）扩展档次；

3）主档次；

4）高档次（定义在 FRExts 修正案中）。

图 4. 1 描述了 H. 264 各种档次的编码工具。图 4. 1 示出了既有公共编码部分又有特殊编码部分的档次。所有档次的公共部分的一部分是：

- I 分片（帧内编码分片）：编码分片只使用来自相同分片内解码样值的预测值；
- P 分片（预测编码分片）：编码分片使用以前解码的参考帧的帧间预测值，至多用到一个运动矢量和参考索引来预测每个块的样值；
- 用作熵编码的 CAVLC（基于上下文的自适应变长编码）。

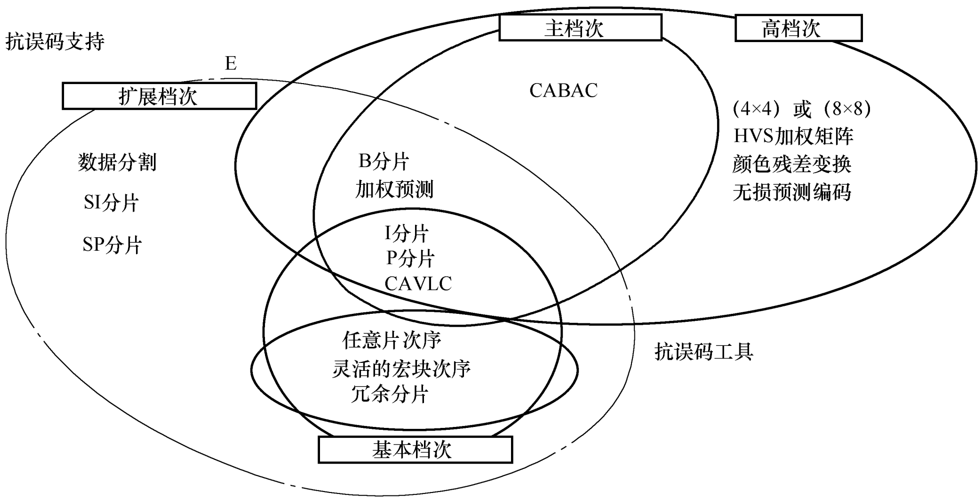


图 4. 1　H. 264 的不同档次以及不同编码工具在档次内的分布［H23］（200 Elsevier）

表 4. 1 列出了 H. 264 档次和每种应用的重要需求。

表 4. 1　H. 264 档次与每种应用的重要需求［B8］

应　　用	需　　求	H. 264 档次
广播电视	编码效率，可靠性（经过噪声可控的信道），隔行视频，低复杂度解码器	主档次
流视频	编码效率，可靠性（经过不可控的分组网络信道），可伸缩性	扩展档次
视频存储和回放	编码效率，隔行视频，低复杂度编码器和解码器	主档次
视频会议	编码效率，可靠性，低时延，低复杂度编码器和解码器	基本档次
移动视频	编码效率，可靠性，低时延，低复杂度编码器和解码，低功耗	基本档次
工作室分布	无损或近无损，隔行视频，有效的转码	主档次，高档次

1. 基本档次

主档次包括的工具列表中有 I（帧内编码）和 P（预测编码）分片编码，灵活宏块排序的强化抗误码工具、任意分片和冗余分片。它也支持 CAVLC（基于上下文的自适应变长编码）。基本档次的设计目的是用在低时延场合下、对处理能力要求低的应用、以及高的丢包环境中。在所有的三个档次中该档次编码效率最低。

- 灵活的宏块顺序（FMO，Flexible Macro- Block Order）：宏块可以不必处于光栅扫描顺序中。该映射将宏块指定给一个分片组；
- 任意分片顺序（ASO，Arbitrary Slice Order）：一帧的一个分片的第一个宏

块的地址可能小于同一编码帧某个其他以前的分片的第一个宏块的地址；

- 冗余分片（RS，Redundant Slice）：与同一分片前面解码的数据相比，该分片属于由相同或不同编码速率获得的冗余编码数据。

2. 主档次

主档次（MP）中包括的编码工具有 I、P 和 B（双向预测编码）分片，隔行编码、CAVLC 和 CABAC（基于上下文的自适应二进制算术编码）。不被主档次支持的工具有抗误码工具、数据分割和 SI（switched intra coded，切换帧内编码的）和 SP（切换预测编码的）分片。这个档次旨在实现最高可能的编码效率。

- B 分片（双向预测编码分片）：编码分片利用前面解码的参考帧的预测值，至多使用两个运动矢量和参考索引来预测每个块的样值；
- 加权预测：通过对运动补偿预测的 P 或 B 分片数据进行加权而进行的缩放操作；
- CABAC（基于上下文的自适应二进制编码）进行的熵编码。

3. 扩展档次

扩展档次（XP）具有基本档次中包括的所有工具。如图 4.1 所示，这个档次也包括 B、SP 和 SI 分片，数据分割、隔行的帧和场编码、图像自适应帧/场编码和宏块自适应帧/场编码。该档次提供比基本档次更好的编码效率。这些附加的工具导致了复杂度的提高。

- 包括基本档次的所有部分：灵活的宏块顺序，任意分片顺序，冗余分片；
- SP 分片：用于在视频流之间有效切换的特殊编码分片，类似于 P 分片的编码；
- SI 分片：切换分片，类似于一个 I 分片的编码；
- 数据分割：编码数据放在不同的数据部分，每部分可被放在不同的层单元内；
- B 分片；
- 加权预测。

4. FRExts 修订案定义的高档次

2004 年 9 月，H.264/MPEG-4 AVC 视频编码标准的第一个修订案发布了［H18］。一组新的编码工具集作为这个修订案的一部分面世了。它们叫作“保真度范围扩展（FIdelity Range Extensions，FRExts）”。发布 FRExts 的目的是为较高保真度材料的编码效率实现明显的改善。FRExts 工具的应用领域包括专业的电影制作、视频制作和高清电视/DVD。

- 包括主档次的所有部分：B 分片、加权预测、CABAC；
- 自适应变换块大小：亮度样值 4×4 或 8×8 的整数变换；
- 量化缩放矩阵：在量化过程按照与变换系数相关的特定频率进行不同的缩放来优化质量。

FRExts 修订案定义了四种新的档次（见图 4. 2）［H11］：

- 高档次（HP）支持 8 比特 4∶2∶0 采样的高分辨率视频应用；
- 高 10 档次（Hi10P）支持 4∶2∶0 采样、最高每样值 10bit 的表示精度；
- 高 4∶2∶2（Hi422P）支持最高 4∶2∶2 的色度采样和最高每样值 10bit 的表示精度；
- 高 4∶4∶4（Hi444P）支持最高 4∶4∶4 的色度采样和最高每样值 12bit 的表示精度，以及编码 RGB 信号的整数颜色残差变换。

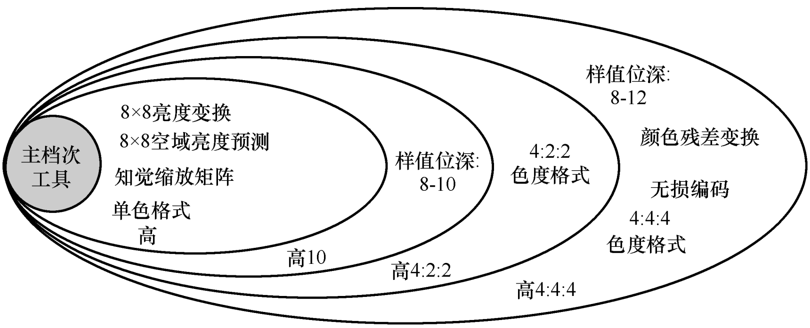


图 4. 2　FRExts 引入的工具以及在新的高档次下的分类［H18］（2005 IEEE）

表 4. 2　高档次与 FRExts 引入的相应编码工具的比较［H11］（2004 SPIE）

编 码 工 具	高档次	高 10 档次	高 4∶2∶2 档次	高 4∶4∶4 档次
主档次工具	×	×	×	×
4∶2∶0 色度格式	×	×	×	×
8bit 样值位深	×	×	×	×
8 ×8 对 4 ×4 变换自适应性	×	×	×	×
量化缩放矩阵	×	×	×	×
Cb 和 Cr 各自的量化参数（QP）控制	×	×	×	×
单色视频格式	×	×	×	×
9 和 10bit 样值位深		×	×	×
4∶2∶2 色度格式			×	×
11 和 12bit 的样值位深				×
4∶4∶4 色度格式				×
颜色残差变换				×
无损预测编码				×

这四个档次进一步建立在以前主档次的设计之上。表 4. 2 提供了 FRExts 引入的高档次的比较，列出了不同的编码工具，以及哪个工具用于哪个档次。所有的高档次包括以下三个编码效率的性能增强措施：

- 在 8 ×8 和 4 ×4 变换块大小之间进行自适应的宏块级切换；
- 在 FRExts 中引入 8 ×8 变换背后的原因是高保真度视频要求保持细节和纹理。为了实现这一目的，就需要用到较大的基函数。然而，较小的变换如 4 ×4 降低了振铃效应和计算复杂度。编码器在 4 ×4 和 8 ×8 之间自适应地选择。

变换选择过程受限于下列条件。

— 如果一个帧间编码宏块具有小于 8 ×8 的子部分（也就是说，4 ×8，8 ×4，4 ×4），那么就必须使用 4 ×4 变换。

— 如果一个帧内编码宏块用 8 ×8 空间预测进行预测，那么只用 8 ×8 变换。

- 编码器规定的基于知觉的量化缩放矩阵。

编码器可以根据特定的频率规定一个缩放因数的矩阵，该频率与用在解码器逆量化的变换系数有关。此举使得按照人类视觉系统的敏感性进行优化成为可能，即对于高频变换系数的编码误差较不敏感［B3］。

- 编码器规定的对每个色度分量量化参数进行独立的控制。

4.2.2　H.264 的级别

图像大小和帧率是影响实现编/译码所需处理能力和存储空间的主要因素。表 4.2 提供了 FRExts 引入的各个高档次的比较，列出了档次中用到的不同的编码工具［H11］。H.264/AVC 定义了 16 种不同的级别，主要受限于图像大小和帧率。级别也对参考帧数以及可用的最大压缩比特率提供约束。每个级别定义了比特流的上界和解码器能力的下界。一个档次和级别可结合起来定义一致点（原文：conformance points——译者注）。这些点表示具有类似功能需求的应用可以协同操作的地方［H8］。为了主要解决 3G 无线网络环境的需求，在 FRExt 修订案中加入“1b”级别。FRExt 档次对于要求更高的高保真度应用在提高比特率能力方面进行了规定。

4.3　H.264 编码器

图 4.3 描述了 H.264 编码器的示意图。像多数其他以往的编译码系统一样，H.264 基于宏块和运动补偿进行工作。视频由一系列图像帧形成。每个图片帧就是一幅图像，一幅图像被细分成块。在 H.264 中，块尺寸可以改变。编码器对于给定图像帧的宏块可以进行帧内编码或帧间编码。帧内编码帧的编码和解码是独立的。它们不需要任何参考帧。因而，它们提供了编码序列的访问点，解码可以从该访问点开始。H.264 在帧内编码中使用 9 种空间预测模式来降低图像源信号中的空间冗余。本节解释了这些模式。帧间编码用到来自某些以前解码帧的一个给定块的预测值。帧间编码的目的是通过利用运动矢量来降低时间冗余。运动矢量给出了一个特定块从当前帧到下一帧的运动方向。然后，得到的预测残差要进行变换来消除块内的空间相关性。接着对这样获得的变换系数进行量化。从帧间预测或帧内模式

得到的运动矢量与量化后的变换系数信息组合在一起。然后对它们用熵编码方法进行编码，例如基于上下文的自适应变长码（CAVLC）或基于上下文的自适应二进制算术编码（CABAC）[H23]。

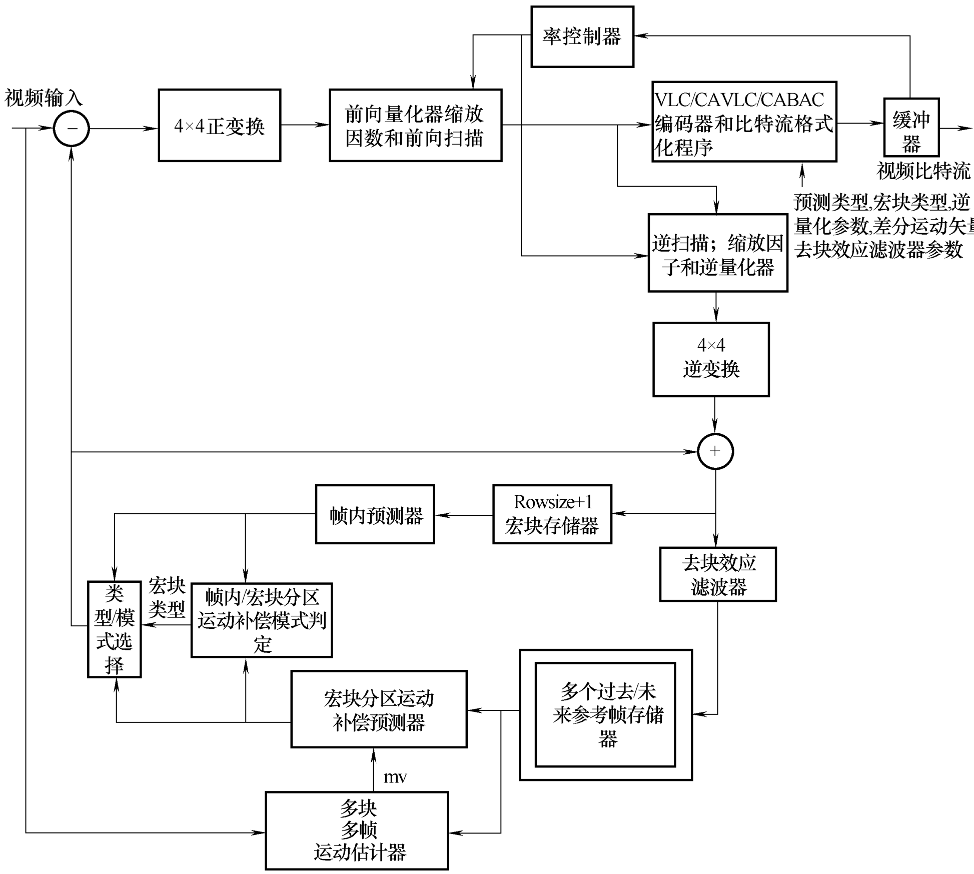


图 4.3　H. 264 编码器框图 [H13]（2004 Elsevier）

H. 264 编码器内部有一个本地解码器。这个解码器执行逆量化和逆变换操作来获得空间域的残差信号。预测信号加上残差信号即可重建输入帧。这个输入帧馈送给去块效应滤波器来去除块边界上的块效应现象。然后，去块效应滤波器的输出馈送给帧间/帧内预测块来生成预测信号。

H. 264 所用的各种编码工具在这一部分解释。

4.4　帧内预测

帧内预测利用像素间的空间相关性，有 3 种定义的基本类型：

- 对于 16×16 完整的亮度或相应大小的色度宏块的预测；

- 对于 FRExts 定义的档次中 8×8 亮度块的预测；
- 4×4 亮度预测。

帧内预测使用来自同一幅图像的宏块进行预测。对于亮度分量，使用两种类型的预测方案。这两个方案可以称为 INTRA_4×4 和 INTRA_16×16［H8］。在 INTRA_4×4 中，一个大小为 16×16 的宏块分为 16 个 4×4 的子块。帧内预测方案单独地运用到这些 4×4 的子块上。总共支持 9 种不同的预测模式如图 4.4 所示［H47］。

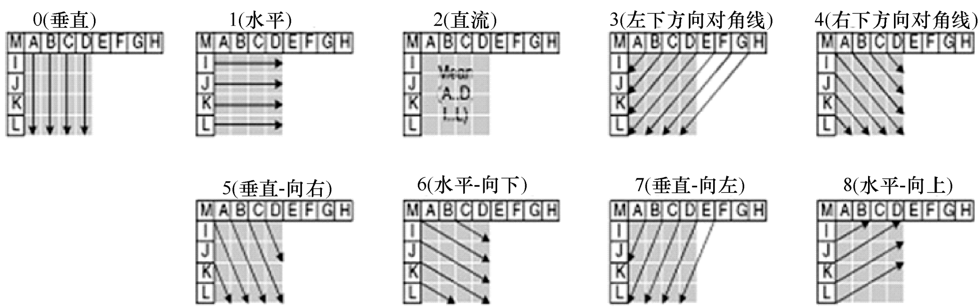


图 4.4　H.264 中 4×4 亮度预测（帧内预测）模式（2010 Wiley）

在模式 0 中，宏块样值用上面相邻的样值进行预测。在模式 1 中，宏块样值用左侧相邻的样值进行预测。在模式 2 中，所有相邻样值的均值用于预测。模式 3 沿左下方向对角线进行预测。模式 4 沿右下方向对角线进行预测。模式 5 是垂直向右方向。模式 6 是水平向下方向。模式 7 是垂直向左方向。模式 8 是水平向上方向。预测值是利用从 A 到 M 的预测值的加权平均计算得到。

在 FRExt 档次中，可以选用 8×8 亮度预测，基本上利用与 4×4 预测相同的概念。8×8 亮度预测针对 8×8 大小的块进行，并且利用预测器的低通滤波器来改善预测性能。

对于亮度分量的 16×16 帧内预测，用到 4 种模式（见图 4.5）。模式 0（垂直）、模式 1（水平）和模式 2（直流）这 3 种模式类似于 4×4 块的预测模式。在第 4 种模式中，相邻样值拟合了线性平面函数。

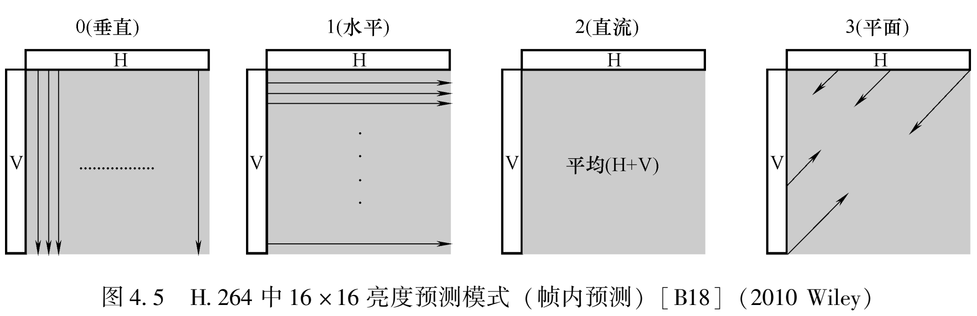


图 4.5　H.264 中 16×16 亮度预测模式（帧内预测）［B18］（2010 Wiley）

色度宏块由相邻色度样值进行预测。用于色度块的 4 种预测模式类似于 16×16 亮度预测模式。对于色度宏块，预测模式的序号有所不同：模式 0 是直流，模式 1 是水平，模式 2 是垂直，模式 3 是平面。色度预测的块大小取决于采样格式。对于 4∶2∶0 格式，选择 8×8 大小的色度块。对于 4∶2∶2 格式，选择 8×16 大小的色度块。对于 4∶4∶4 格式，选择 16×16 大小的色度块［H23］。

4.5　帧间预测

帧间预测用于充分利用视频序列中的时间冗余。时间相关性通过使用运动估计和补偿算法的帧间预测得以降低［H23］。一幅图像分成多个宏块；每个 16×16 的宏块进一步被分割成 16×16、16×8、8×16、8×8 大小的块。一个 8×8 的亚宏块可被进一步分成 8×4、4×8、4×4 大小的块。图 4.6 所示为一个宏块和一个亚宏块的分割情况［H7］。输入视频的特点决定了块的大小。较小的块大小保证了较小的残差数据；然而，较小的块大小也意味着更多的运动矢量和因此带来编码这些运动矢量所需的更多的比特数。［B18，H48］。H.264 支持直到 1/4 像素分辨率的运动矢量（见后面图 4.9）。

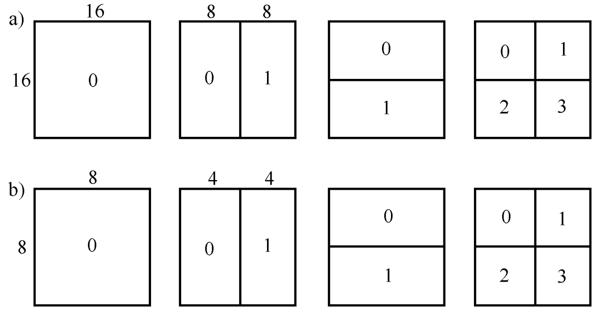


图 4.6　H.264 中帧间预测的宏块分割［H3］（2003 IEEE）
a)（L-R）16×16、8×16、16×8、8×8 大小的块　b)（L-R）8×8、4×8、8×4、4×4 大小的块

4.6　P 分片中宏块的帧间预测

H.264 视频编译码系统使用基于块的运动补偿，每个主要视频编码标准采纳这个同样的原则。H.264 提供某些重要的区别，如块尺寸小至 4×4 和亮度分量中亚像素运动矢量细化到 1/4 像素的精度［H44］。

这些分割和亚分割在每个宏块内产生大量的可能组合。这种把宏块分割成大小不同的运动补偿宏块的方法叫作树形结构运动补偿。分割的选择很重要，因为它对于压缩性能具有显著影响。一般而言，对于一帧中的同源区域大的分割尺寸是合适

的，对于非同源区域小的分割尺寸可能是有益的。选择大的分割尺寸（16×16、16×8、8×16 和 8×8）意味着只需要不多的比特表示运动矢量和分割类型的选择；然而，运动补偿残差可能包含帧区域中丰富细节的显著的能量。选择小的分割尺寸（8×4、4×8 和 4×4）在运动补偿后可能给出较低的能量残差但需要大量的比特来表示运动矢量和分割的选择［H44］。

宏块（Cr 和 Cb）的每个色度分量的分辨率是亮度分量的一半。每个色度块按照与亮度分量相同的方式进行分割，除了分割尺寸正好是水平和垂直分辨率的一半之外。亮度的 8×16 分割相应于色度的 4×8 分割［H44］。

4.7　亚像素运动矢量

帧内编码宏块的每个部分由参考帧中相同大小的区域来预测。两个区域之间的偏移，即运动矢量，对于亮度分量具有 1/4 像素的精度。在亚像素运动矢量预测中，参考帧没有亚像素位置，所以有必要从临近的图像样值中用内插法来创建它们［H44］。

亚像素运动补偿能提供明显好于整数像素补偿的压缩性能，但是增加了复杂度。1/4 像素精度性能优于半像素精度［H44］。在亮度分量中，首先生成半像素位置上的亚像素样值，而且用一个六抽头有限脉冲响应滤波器从相邻的整数像素样值中进行内插。图 4.7 和 4.8 示出了用来获得半像素亮度样值的预测值的六抽头滤波器的权重［B18］（1，－5，20，20，－5，1）/32，以获得更尖锐的亚像素运动补偿。1/4 像素运动估计通过半像素值的线性内插获得，以节约处理功率。

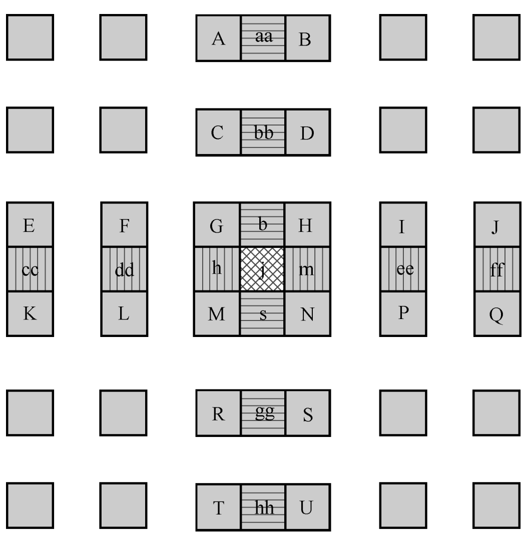


图 4.7　亮度半像素位置上的内插（2010 Wiley）

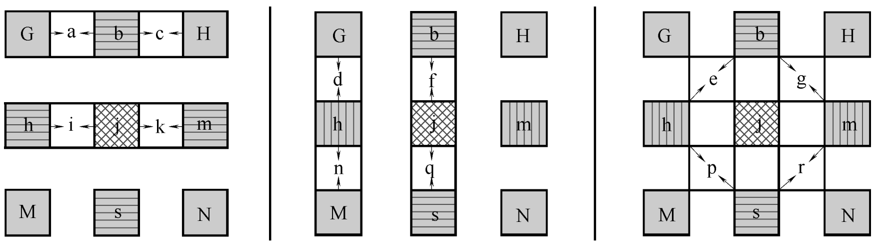


图 4. 8　亮度 1/4 像素位置上的内插（2010 Wiley）

这意味着每个半像素样值是 6 个相邻整数像素样值的加权和。一旦可以得到所有的像素样值，每个 1/4 像素样值就可以在相邻的半像素或整数像素的样值上用双线性内插来产生［H44］。图 4. 9 给出了一个整数像素和亚像素预测的例子。当前帧内 4 ×4 亚分割区域可以从参考帧的相邻区域预测。如果运动矢量的水平和垂直分量都是整数，那么参考块里的相关样值确实存在，如图 4. 9b 所示。如果两个矢量分量或者其中之一是分数值，那么预测样值就要从参考帧中的相邻样值中内插。

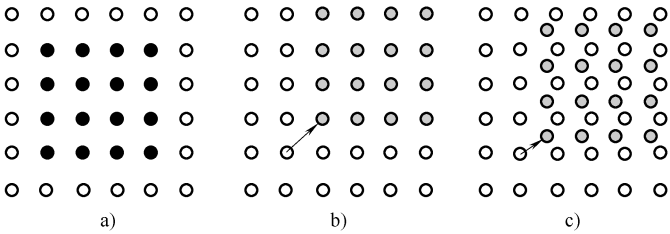


图 4. 9　整数和亚像素运动矢量；H. 264 支持高达 1/4 像素分辨率的运动矢量［B18］（2010 Wiley）

用作帧间预测的参考帧是以前解码的帧，并存储在帧缓冲器中。H. 264 支持使用多帧作为参考帧。这是通过使用附加的与运动矢量一起传输的帧参考参数来实现的。图 4. 10 中参数 t 和 d 就是帧参考参数。

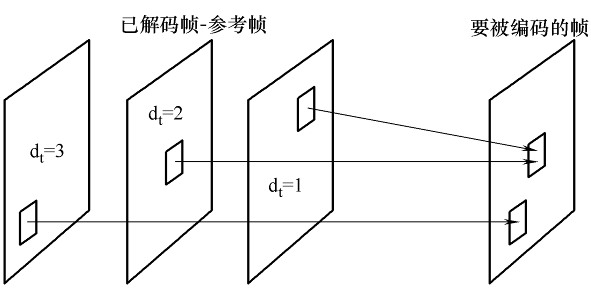


图 4. 10　多参考帧运动补偿预测［H8］（2004 IEEE）

4.8 变换与量化

对一个残差样值块用 4×4 或 8×8 整数变换进行变换处理，这是一种离散余弦变换（DCT）的近似形式［B2］。离散余弦变换输出一组系数，每个系数是标准基函数的加权值。加权基函数被组合之后就会重建残差样值块。

变换的输出系数要按照量化参数（QP）进行量化。量化降低了变换系数的精度。量化参数是每个变换系数除以的整数值。

在预测误差信号中存在高度的空间冗余。H.264 执行基于块的变换以降低这种空间冗余［H23］。以前的标准 MPEG-1 和 MPEG-2 使用二维离散余弦变换（DCT）来实现大小为 8×8 块变换编码的目的［H8］。H.264 使用整数变换而不是 DCT。这些变换的大小是 4×4［H8］。H.264 中使用较小块尺寸的优势如下所述：

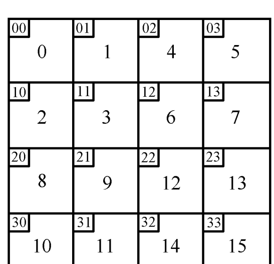
- 变换尺寸的降低使得编码器能够更好地让预测误差编码适应于运动对象的边界，而且用最小的运动补偿块尺寸匹配变换块尺寸；
- 较小的变换块尺寸导致振铃效应的明显降低；
- 4×4 变换具有不需要乘法的优势。

H.264 使用了分级的变换结构，在此结构中亮度信号相邻 4×4 变换的（见图 4.11a）直流系数被分成 4×4 的块，而且由哈达玛（Hadamard）变换进行再次变换（见图 4.11b）。如图 4.11a 所示，第一个变换（图 4.11f 的矩阵 H1）应用于所有亮度分量（Y）预测误差块的样值以及所有的色度分量（Cb 和 Cr）块。对于主要是平坦像素值的块，在相邻块的直流系数之间存在显著的相关性。因而，标准对于亮度直流系数只为 16×16 帧内模式规定了 4×4 的哈达玛变换（图 4.11f 中的

$$[Y]=\begin{bmatrix}1&1&1&1\\2&1&-1&-2\\1&-1&-1&1\\1&-2&-2&-1\end{bmatrix}\begin{bmatrix}x_{00}&x_{01}&x_{02}&x_{03}\\x_{10}&x_{11}&x_{12}&x_{13}\\x_{20}&x_{21}&x_{22}&x_{23}\\x_{30}&x_{31}&x_{32}&x_{33}\end{bmatrix}\begin{bmatrix}1&2&1&1\\1&1&-1&-2\\1&-1&-1&2\\1&-2&1&-1\end{bmatrix}$$

$$[Y]=[H_1][X][H_1]^{\mathrm{T}}$$

a)

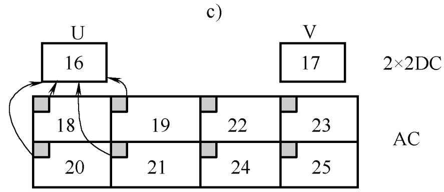


b)

图 4.11　a）矩阵 H1（f）用于原始（4×4）像素且不用于（4×4）直流系数（b）［H7］
b）16 个 4×4 亮度块的直流系数［H7］（2003 IEEE）

$$\left(\begin{bmatrix}1&1&1&1\\1&1&-1&-1\\1&-1&-1&1\\1&-1&1&-1\end{bmatrix}\begin{bmatrix}x_{D00}&x_{D01}&x_{D02}&x_{D03}\\x_{D10}&x_{D11}&x_{D12}&x_{D13}\\x_{D20}&x_{D21}&x_{D22}&x_{D23}\\x_{D30}&x_{D31}&x_{D32}&x_{D33}\end{bmatrix}\begin{bmatrix}1&1&1&1\\1&1&-1&-1\\1&-1&-1&1\\1&-1&1&-1\end{bmatrix}\right)//2$$

c)



d)

$$\begin{bmatrix}1&1\\1&-1\end{bmatrix}\begin{bmatrix}DC_{00}&DC_{01}\\DC_{10}&DC_{11}\end{bmatrix}\begin{bmatrix}1&1\\1&-1\end{bmatrix}$$

e)

$$H_1=\begin{bmatrix}1&1&1&1\\2&1&-1&-2\\1&-1&-1&1\\1&-2&2&-1\end{bmatrix}\qquad H_2=\begin{bmatrix}1&1&1&1\\1&1&-1&-1\\1&-1&-1&1\\1&-1&1&-1\end{bmatrix}\qquad H_3=\begin{bmatrix}1&1\\1&-1\end{bmatrix}$$

f)

图 4.11（续）　c）矩阵 H2（f）（4×4 哈达玛变换）用于（4×4）直流系数［H7］（2003 IEEE）
d）来自两个（8×8）色度块的直流系数［H7］（2003 IEEE）
e）矩阵 H3（f）（2×2 哈达玛变换）用于色度直流系数（d）［H7］（2003 IEEE）
f）用在 H.264 中三个变换的矩阵 H1、H2 和 H3（2004 IEEE）

矩阵 H2)，对于色度直流系数则规定了 2×2 的哈达玛变换，如图 4.11d 和 e (图 4.11f中的矩阵 H3) 所示。

4.9　环路去块效应滤波器

环路去块效应滤波器（见图 4.3）用来去除因基于块编码模式而产生的块效应。帧内预测或帧间预测之后的变换作用在块上；然后变换系数要经过量化。块效应要归咎于这些基于块的操作，它通过使用环路去块效应滤波器加以消除。它减轻了块边界的瑕疵，阻止了累积误差的传播。然而滤波器的存在增加了系统的复杂度［H50］。图 4.12 描述了一个具有 16 个 4×4 子块的宏块及其边界。

如图 4.12 所示，亮度去块效应滤波过程在 16 个样值边沿上执行——由实线所示。色度去块效应滤波过程在 8 个样值边沿上执行——由虚线所示。

H.264 分别在编码器和解码器中逆变换之后使用了自适应的环路去块效应滤波器。该滤波器作用于每个解码的宏块以降低块失真［B18］。在编码器中，在重建和存储用于未来预测的宏块之前，逆变换之后就要使用去块效应滤波器；在解码器中，在重建和显示宏块之前使用该滤波器。滤波器使得块边沿变得平滑，改善了解码帧的显示质量。因为滤波之后的帧比未经滤波斑斑驳驳的帧更忠实地再生了原始帧，所以滤波帧用于未来帧的运动补偿一般来说都可以改善压缩性能。

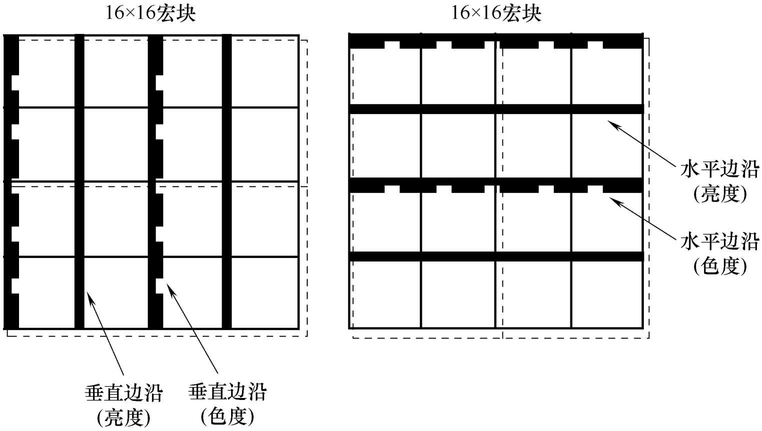


图4.12　欲被滤波的宏块边界（亮度边界以实线所示，色度边界以虚线所示）

去块效应滤波器的操作可被分成三个主要的步骤，分别是滤波器强度计算、滤波器判决和滤波器实现。

4.9.1　滤波器强度

滤波器强度也就是滤波量，在参数边界强度的帮助下进行计算。滤波器的边界强度取决于当前的量化器、宏块类型、运动矢量、跨越边界图像样值的梯度、以及其他参数。对于4×4相邻块之间的每个边沿，得出边界强度；而且对于每个边沿，赋予边界强度参数一个0~4之间的整数值。在可能存在明显块失真的地方诸如帧内编码的宏块边界或者在包含编码系数的块边界上，滤波器是比较强的［H22］。

除了分片边界的边沿外，滤波过程作用于宏块内块的垂直和水平边沿。图4.13表明，宏块级的滤波顺序是对亮度分量的4个垂直边界以亮度边沿（VLE，Vertical Luminance Edge）VLE1、VLE2、VLE3和VLE4的顺序进行滤波的。被滤波的亮度分量水平边沿（HLE，Horizontal Luma Edge）是HLE1、HLE2、HLE3和HLE4。最后，垂直色度边沿（VCE，Vertical Chrominance Edge）VCE1、VCE2和水平分量（HCE，Horizontal Chrominance Edge）HCE1、HCE2再进行分别滤波。并且，滤波器也有可能改变其强度或者关闭滤波器。

图4.14示出了滤波操作对边界两侧的4个样值的影响。相邻块的垂直边沿或水平边沿的4个样值是p0、p1、p2、p3和q0、q1、q2、q3［H22］。

H.264在以下3个级别上自适应地使用去块效应过程：

- 在分片级——全局滤波强度调整到视频序列的独有特征；
- 在块边沿级——去块效应滤波器判决基于块的帧间或帧内预测、运动差分和两个相关块编码残差的出现；
- 在样值级——区别块效应和图像真实的边沿是很重要的。真实边沿不应该被滤除掉。因而样值级别上对于去块操作的判决就变得很重要了。

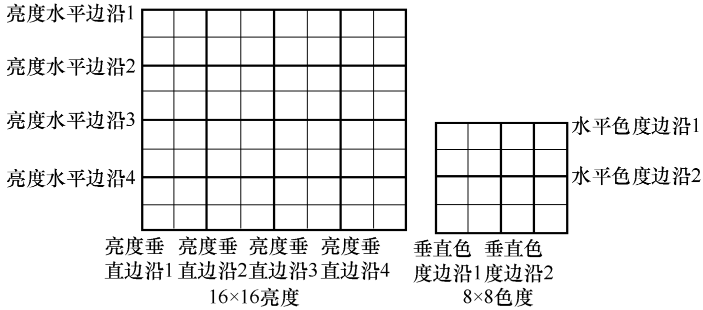


图 4. 13　宏块的边沿滤波顺序［H22］（2006 IEEE）

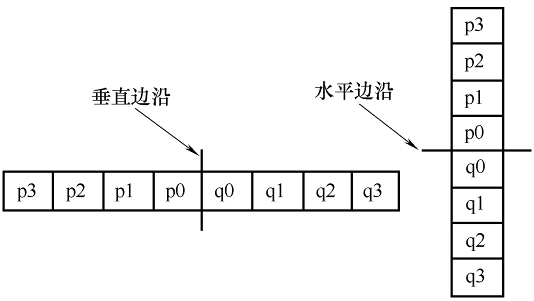


图 4. 14　邻近水平和垂直边界的像素［H22］（2006 IEEE）

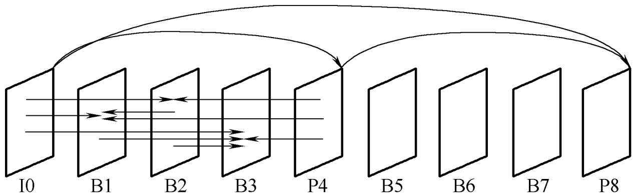
4. 10　B 分片与自适应加权预测

过去和未来帧都用作参考的双向预测可能在改善时域预测方面是非常有用的。H. 264 中的双向预测使用了多个参考帧。图 4. 15a 示出了从多个参考帧进行的双向预测。H. 264 之前带有 B 帧的标准使用双向模式，限于容许前面和后面预测信号的组合。在以前的标准里，一个预测信号从后面的帧中获得，再一个预测信号从前面的帧中获得，另外一个从两个运动补偿预测信号的线性平均信号中获得。

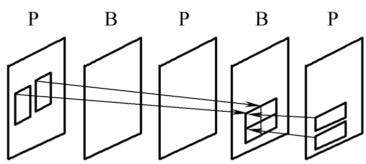
H. 264 支持前向/后向预测对，也支持前向/前向和后向/后向预测对［H23］。图 4. 15b 和 c 分别描述了双向预测和多参考帧的方案，因而宏块被预测为多个参考信号的线性组合，组合权重如 4. 1 所示。在场景改变之前，考虑两个做预测的前向参考帧对于一个区域的运动补偿预测是有益的。而在场景改变之后，考虑两个后向参考帧是有益的。H. 264 也允许双向预测编码的分片可被用作其他帧帧间编码的参考。除 H. 264 之外，所有现存标准都对于参考帧赋予相同的权重。参考信号的等值权重是平均分配的，于是获得预测信号。H. 264 也用到加权预测［H23］。它可被用作 P 分片或 B 分片的宏块。不同的权重可被指定给两个不同的参考信号，预测信号计算如下：

$$p = w_1 \times r_1 + w_2 \times r_2 \tag{4.1}$$

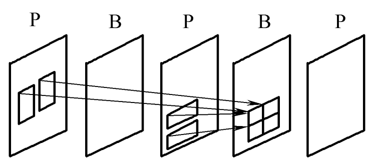
式中，p 为预测信号；r_1 和 r_2 为参考信号；w_1 和 w_2 为预测权重。



a)



b)



c)

图 4.15 双向预测

a）双向预测 b）双向模式（带有过去和未来宏块预测信号的线性组合）

c）多参考帧模式（带有两个过去宏块预测信号的线性组合）

4.11 熵编码

视频编码过程的最后一步是熵编码。熵编码基于如下原则：将较短码字赋予出现概率较高的符号，即那个较长码字赋予出现概率较低的符号。要进行熵编码的部分参数包括残差数据的变换系数、运动矢量以及其他编码器信息。标准已经采纳了两种类型的熵编码。第一种方法以通用变长编码（UVLC，Universal Variable Length Coding）和自适应变长编码（CAVLC，Context Adaptive Variable-Length Coding）为代表。第二种方法以基于上下文自适应二进制算术编码为（CABAC，Context-Based Adaptive Binary Arithmetic Coding）代表。

H.264 码流或 H.264 文件包括一系列编码符号。这些符号组成语法，包括参数、标识符和、定界码、预测类型、差分编码的运动矢量和变换系数。H.264/AVC 标准对于编码符号规定了几种方法，即把每个符号转换成二进制形式作为码流的一部分进行传输或存储。这些方法如下：

定长码：符号转换成一个具有规定长度（$n$bit）的二进制码，每个码字长度固定。在定长编码方法中，只有对于大块的数据才可能实现数据压缩，任何超越概率

总数的对数的压缩都会以一个有限的概率遭致失败。

指数哥伦布变长码［B18］：符号表示为一个比特数不同的指数哥伦布码字。总的说来，较短的指数哥伦布码字赋予出现较为频繁的符号。

CAVLC（上下文自适应变长编码）：上下文自适应变长编码使用上下文自适应性，是一种为变换系数专门设计的方法。其中，选择不同的变长码集取决于最近编码系数的统计分布。

在预测、变换和量化之后，块一般都是稀疏的，经常包括的大部分是零。CAVLC 利用游程编码来紧凑地表示连续的零串。块扫描之后最高的非零系数常常是 ±1 的序列，且 CAVLC 给出高频系数个数。相邻块非零系数的个数具有相关性。系数个数采用查表法进行编码，所查表的选择取决于相邻块非零系数的个数。非零系数的幅值在扫描矩阵的起始位置（靠近直流系数）趋于增大，而在较高频率附近趋于减小。

H. 264 基于上下文特点用变长编码来将一个符号与一个码字进行匹配。除了残差数据之外，所有的语法元素都用指数哥伦布码进行编码［H23］。残差数据用 CAVLC 进行编码。

经过变换和量化之后，系数的幅值是 0 或 ±1 的概率是非常高的［H23］。CAVLC 对这些值区别对待。它对 0 和 ±1 编码它们的个数。对于其他值，是编码它们的值。

1）没有块末尾，而是系数的个数要进行解码；

2）向后扫描系数，上下文依据编码系数而建立；

3）变换系数以下列元素进行编码：非零系数个数、所有非零系数的幅值和符号、在最后非零系数之前的零的总数、以及在最后非零系数之前的连零的个数；

4）所用的变长码表基于相邻块系数个数进行自适应地选择。

CABAC（上下文自适应的二进制算术编码）：上下文自适应二进制算术编码［B18］是一种算术编码方法，其中的概率模型基于前面的编码统计数据进行更新。CABAC 在主档次和高档次是一种可选的熵编码模式。CABAC 通过一些措施实现很好的压缩性能：

（a）按照元素的上下文来为每个语法元素选择概率模型。

（b）基于局部统计数据调整概率估计。

（c）使用算术编码而不是变长编码。

图 4. 16 示出了 CABAC 的总图［H11］。编码一个数据符号包括如下几步：

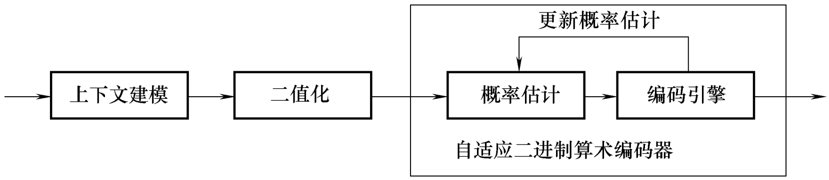


图 4. 16　CABAC 的示意框图［H11］（2004 SPIE）

二值化：CABAC 使用二进制算术编码，这意味着只对二进制判决符（1 或 0）进行编码。非二进制的数值符号要先于算术编码之前转换成一个二进制码。

上下文模型选择："上下文模型"是二值化符号的一个或多个比特的概率模型，它依据最近编码的数据符号的统计分布从可用的模型中进行选择。

算术编码：算术编码器按照所选的概率模型对每个 bin 进行编码。注意，只有两个子范围值 1 或 0。

概率更新：所选的上下文模型基于实际编码值进行更新。

1）对于大多数符号使用自适应概率模型；

2）通过使用上下文利用符号相关性；

3）通过在二进制序列中的位置区别二进制判决符；

4）通过查表法实现概率估计。

4.12 H.264 解码器

H.264 解码器在操作上的允许类似于 H.264 编码器的本地解码器。解码完全就是编码的相反过程。视频解码器接收到 H.264 的压缩码流，对语法元素进行解码，抽取量化的变换系数、预测信息等。这些数据用于重建视频序列。在比特流上进行熵解码（CABAC 或 CAVLC）以获得变换系数。然后，这些系数进行逆扫描和逆量化（量化的变换系数乘以量化参数。量化参数是一个整数值）。这一步给出了变换域上的残差数据块。再进行逆变换来获得像素域上的数据。变换系数被再次缩放之后，逆变换将标准基函数组合起来，被再缩放系数加权，来重建每一块的残差数据。这些块组合到一起就形成了残差数据宏块。对于每个宏块，解码器进行与编码器创建的完全相同的预测。依据帧间预测或帧内预测，把适当的预测信号加到残差信号上去。这个预测块加到残差块上来重建视频帧。这些重建帧在存储起来用作未来的预测之前或者被显示之前，都要进行去块效应处理。

4.13 H.264 的一些应用

RGB Spectrum（一家美国公司——译者注）的 DSx H.264 编译码系统［H52］能提供极好的图像质量，能无瑕疵地编码实况视频和图形并把他们通过加密的 SATCOM 卫星通信从太平洋导弹测试场传输到 Colorado Spring、Colorado 和 Huntsville、Alabama 和其他华盛顿特区以外的地方。然后，存储该码流以备进一步的系统性能、目标获取精度和目标毁坏性的后续测试分析。

DSx 编译码系统的优势是，使用了 H.264 高档次这一高端的压缩技术来传送相比于传统主档次和基本档次更好的图像质量。这种优越的压缩能力保持了包含在导弹测试影像中的复杂细节。

DSx 编译码系统以每秒 30 帧的实时速率、高达 1920 × 1080 的分辨率对视频和图形影像进行编码。操作员在一台 PC 机上使用基于网络的图形用户界面（GUI）对 DSx 编译码系统进行控制。该编译码系统提供带有标题和精度时间码的屏幕显示，这是嵌入在影像流之内的。码流带宽从 64kbit/s ~ 10Mbit/s 可调，取决于信号类型。

4.14　总结

本章概括了 H. 264 编译码系统的编码工具。H. 264/AVC 工程的意图是建立一个标准，使其能够在大大低于以前标准的比特率（即 MPEG-2、H. 263 或 MPEG-4 第 2 部分的一半或更低的比特率）上提供好的视频质量，而没有增加设计的复杂度以避免它不实用或实现代价过分高昂。H. 264 标准可以看作是一个“标准之家”，其中的成员是表 4. 3 所描述的各个档次。H. 264/AVC 的附加部分如可伸缩视频编码（SVC）和多视点视频编码没有讨论。下一章陈述了最新的标准——高效视频编码（HEVC），它在压缩比和视觉质量方面甚至比 H. 264/AVC 更加有效。

表 4. 3　H. 264 定义的级别［H11］

级别号	典型的图像尺寸	典型的帧率	非 FRExt 档次中最大压缩比特率（对 VLC）	典型图像尺寸的最大参考帧数
1	QCIF	15	64kbit/s	4
1b	QCIF	15	128kbit/s	4
1. 1	CIF 或 QCIF	7. 5(CIF0)/30(QCIF)	192kbit/s	2(CIF)/9(QCIF)
1. 2	CIF	15	384kbit/s	6
1. 3	CIF	30	768kbit/s	6
2	CIF	30	2Mbit/s	6
2. 1	HHR(480i 或 576i)	30/25	4Mbit/s	6
2. 2	SD	15	4Mbit/s	5
3	SD	30/25	10Mbit/s	5
3. 1	1280 × 720p	30	14Mbit/s	5
3. 2	1280 × 720p	60	20Mbit/s	4
4	HD 格式（720p 或 1080i）	60p/30i	20Mbit/s	4
4. 1	HD 格式（720p 或 1080i）	60p	50Mbit/s	4
4. 2	1920 × 1080p	60p	50Mbit/s	4
5	2k × 1k	72	135Mbit/s	5
5. 1	2k × 1k 或 4k × 2k	120/30	240Mbit/s	5

半水平分辨率（HHR：Half-horizontal resolution）（2004 SPIE）

4.15 专题研究

P.4.1 在附录E中，JM 7.2、Intel IPP和x264的性能分析和对比中，提供了实现H.264的软件。对于基本档次、主档次和高档次，比较参数是SSIM、PSNR、总的编码时间、和压缩比。附录表明，Intel IPP和x264的编码时间显著低于JM17.2。请使用各种标准测试序列来实现H.264的Intel IPP、x264和JM（最新版本）并获得基本档次、主档次和高档次的类似性能分析。

P.4.2 使用FFmpeg软件重复P.4.1。

P.4.3 对于4:2:2和4:4:4的视频格式重复P.4.1和P.4.2。

P.4.4 请从UTA网址-courses EE5359→theses访问Madhu Peringassery Krishnan的学位论文（Implementation and performance analysis of 2-D order 16 integer transforms in H.264/AVC and AVS-video for high definition video coding）。

H.264和AVS-China中用于SICT和MICT的16阶变换矩阵分别如论文的65页和70页所示。请开发这些变换快速实现的流图。

binDCT-L也实现了。H.264和AVS-China（第3章）实现的时候，对于不同的测试序列在各种量化参数下都嵌入了这些逆变换。请实现这些操作并获得类似与这篇论文所示的那些比较结果。

P.4.5 请开发更高阶的2维ICT（32×32和64×64）。这些是由HEVC提出的（第5章）。请用高清和超高清测试序列把这些嵌入到H.264和AVS China中并进行类似于Madhu Krishnan论文所示那样的性能对比。

P.4.6 在Santosh Muniyappa的学位论文（见UTA网址）中，实现了H.264/AVC中帧内模式选择的复杂度降低算法。使用简单的方向掩模和相邻模式，模式组合数从592减少到至多132（RDO计算）。论文使用了QCIF和CIF格式的视频序列。请将这个复杂度降低算法推广到其他视频格式，诸如4SIF和HD。

P.4.7 从UTA网址访问Amruta Kulkarni的学位论文“Fast inter-prediction mode decisionalgorithm for H.264 video encoder”。由于使用了一种快速的自适应终止模式选择算法，它能够显著降低编码器的实现时间（与JM 17.2参考软件相比），伴随着可忽略的视频质量和极小的比特率提高。她使用了H.264基本档次中QCIF和CIF格式的视频序列。请通过使用快速自适应终止（FAT）算法来确认降低的实现复杂度（使用不同的视频测试序列）。

P.4.8 对主档次和高档次重复P.4.7。注意，在这些档次中也有（双向内插）B帧。这使得在H.264中采用FAT算法甚至更为复杂。

P.4.9 请用QCIF和CIF格式对基本档次将Muniyappa开发的H.264/AVC中用于

帧内模式选择的复杂度降低算法，和 Kulkarni 采用的 FAT 算法（H. 264 视频编码器的快速帧内预测模式判决算法）组合起来，并评价 H. 264/AVC 编码器在实现复杂度方面降低的情况。请在不同的比特率下使用多种测试序列。

P. 4. 10　请用标清和高清序列对主档次和高档次重复 P. 4. 9。

P. 4. 11　请从 UTA 网址访问 Shreyanka Subbarayappa 的学位论文 "Implementation and analysis of directional discrete cosine transform（DDCT）in H. 264 baseline profile" 尽管在 H. 264 中采用方向 DCT 导致编码时间的显著增加，但回报是改善了图像质量（SSIM、PSNR）。请在各种比特率下使用不同的测试序列来实现 H. 264 中的 DDCT。

P. 4. 12　对主档次和高档次重复 P. 4. 11。

P. 4. 13　Shreyanka（见 P. 4. 11）只是在 I 帧中实现了 DDCT。在第 4 章，考虑到不同的方向模式示出了 4 ×4 和 8 ×8 块的基元图像。请开发出这些基元图像。对所有关于（4 ×4）和（8 ×8）DDCT 的方向模式，获得这些基元图像。模式 4、6、7 和把可以从模式 3 和 5 获得，如论文所示。请将这种方法用于获得基元图像。

P. 4. 14　请从 UTA 的网址访问 T. Purushotham 的学位论文 "Low complexity encoder using machine learning"。基于仅仅 4 帧的训练，利用 QCIF 和 CIF 测试序列，基本档次的运动估计时间平均下降 42. 86%，编码时间平均下降 23. 5%。论文只用到帧间亚宏块（8 ×8、8 ×4、4 ×8 和 4 ×4）模式判决。请将这一方法推广到所有的宏块模式，也就是说包括 16 ×16、16 ×8 和 8 ×16 的块。也要基于 100 帧的训练而不是仅仅 4 帧来重复该仿真。

P. 4. 15　请用标清和高清测试序列对主档次和高档次重复 P. 4. 14。

P. 4. 16　Veerla（见附录 F）开发了将 H. 264 和 JPEG 组合到一起的先进图像编码（AIC，Advanced Image Coding）。AIC 与 H. 264 的帧内编码和其他静止图像编码标准进行了比较，这些标准如 JPEG、JPEG-LS、JPEG2000 和 JPEG XR（高清照片）。将 SSIM（附录 C）、PSNR、MSE 和计算时间作为测度，在不同的比特率上用一些测试图像进行了仿真。与其他编码标准相比，AIC 性能良好且降低了实现复杂度。请重复这些仿真（使用一些其他测试图像）。参考文献中提供了各种图像编码标准实现软件（免费软件）的获取途径。这可作为一个小组专题研究，不同的学生来实现不同的标准（JPEG、JPEG-LS、JPEG2000、JPEG-XR、H. 264 帧内编码和微软 HD-photo）。

P. 4. 17　C. Deng 等人对 H. 264/AVC FREXT 加入了进一步的扩展，例如更大的运动矢量搜索范围、更大的宏块和跳过块尺寸以及 I-D DDCT。他们利用高分辨率（HR）（4096 ×2160）视频序列与运动 JPEG2000 进行了性能对比，结果表明，在各种比特率上前者的 PSNR 得到显著改善。请用 HR 测试

序列实现扩展的 H. 264 和运动 JPEG2000 并证实前者具有优越的性能。C. Deng 等，“Performance analysis，parameter selection and extension to H. 264/AVC FREXT for high resolution video coding”，J. VCIR，vol. 22，pp. 687-760，Feb. 2011.

P. 4. 18 通过加入一些额外的特征诸如扩展的块尺寸（直到 64 × 64）、帧内编码中模式相关的方向变换（MDDT）、亮度和色度的高精度滤波、自适应系数扫描、扩展的块尺寸分割、自适应环路滤波、大尺寸整数变换等，Karczewicz 等人提出了一种优于 H. 264/AVC 的混合视频编译码系统。通过在不同的空间分辨率上使用几种测试序列，他们发现，在主观质量和客观指标方面新编译码系统性能均优于传统的 H. 264/AVC。这一些系统也需要在编码器和译码器的复杂度方面有适度的增加。请实现该新编译码系统并获得类似与这篇文章描述的结果。也将 SSIM 作为在所有仿真中的另一个性能指标。使用最新的 H. 264 JM 软件。“A hybrid video coder based on extended macroblock sizes，improved interpolation，and flexible motion representation”，IEEE Trans. CSVT，vol. 20，pp. 1698-1708，Dec. 2010.

P. 4. 19 在文档 JCTVC-G399 r2 中，Bin Li 比较了 HEVC WD4（第 5 章）和 AVC 高档次的压缩性能。他在不同的比特率下基于几个测试序列用 HEVC WD7 和最新的 H. 264 软件 JM 进行了对比。如同以前，性能指标为 SSIM、PSNR、比特率和实现复杂度。瑞士日内瓦第 7 届 JCT-VC 会议，2011 年 11 月 21 ~ 30 日（HEVC 与 AVC 高档次压缩性能对比的工作草案）。

第5章　高效视频编码（HEVC）

摘要：本章给出了HEVC这一最新的标准，引用了与H.264/AVC（第4章）的对比情况。重点是HEVC的概述而不是构成编码器工具和方法的详细描述。最好，列出了许多研究专题挑战有关HEVC的实现和进一步的研究。

关键词：HEVC　JCTVC　统一的帧内预测　编码树单元　预测单元　变换单元　SAO（样值自适应偏移）　系数扫描　HM软件　无损编码

5.1　引言

本章详细说明了由视频编码联合协作团队（JCT-VC）完成的HEVC的制订。

5.2　视频编码联合协作团队

视频编码联合协作团队是一个视频编码专家组，这些专家来自ITU-T第16研究组（VCEG）和ISO/IEC JTC 1/SC 29/WG 11（MPEG），建立该专家组是为了制订新一代的视频编码标准，该标准与目前最先进的AVC标准（ITU-T Rec. H.264 | ISO/IEC 14496-10）相比，将进一步降低高质量视频编码所需数据率的50%。这个编码标准化新方案叫作高效视频编码（HEVC，High Efficiency Video Coding）。在ISO/IEC中，它叫作MPEG-H第2部分。

ITU-T推荐草案H.264 | ISO/IEC 14496-10，通常称为H.264/MPEG-4AVC，H.264/AVC，或MPEG-4第10部分AVC（第4章）已经发展成为一种JVT内部的联合活动。各种视频编码标准的演化如图5.1所示。

JCT-VC由Jens-Rainer Ohm和Gary Sullivan联合主持，他们的联系信息如下提供。

JCT-VC的ITU-T联系信息	会议
Gary Sullivan先生 大会报告起草人，视觉编码 问题6，ITU-T第16研究组 电话：+1 425 703 5308 传真：+1 425 936 7329 Email：garysull@microsoft.com	2013年10月，瑞士日内瓦 2013年7月27日-8月2日，奥地利维也纳 2013年4月20-26日，韩国仁川 2013年1月14-23日，瑞士日内瓦

（续）

JCT-VC 的 ITU-T 联系信息	会议
Thomas WIEGAND 先生 大会报告副起草人，视觉编码 问题 6，ITU-T 第 16 研究组 电话：+49 30 31002 617 传真：+49 30 392 7200 Emmail：thomas. wiegand@ microsoft. com JCT-VCd 的 ISO/IEC 联系信息 Jens-Rainer OHM 先生 大会报告起草人，视觉编码 问题 6，ITU-T 第 16 研究组 电话：+49 241 80 27671 Email：ohm@ ient. rwth-aachen. de	Gary SULLIVAN 先生 大会报告起草人，视觉编码 问题 6，ITU-T 第 16 研究组 电话：+1 425 703 5308 传真：+1 425 936 7329 Email：garysull@ microsoft. com

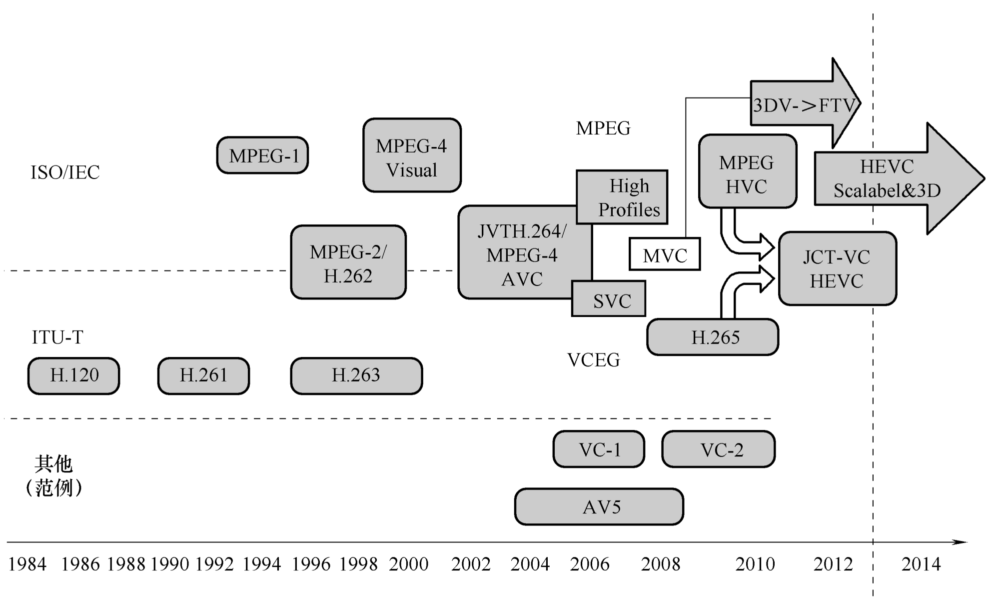


图 5.1　视频编码标准（承蒙美国加利福尼亚州圣塔　克拉拉大学计算机工程系的圣菲利波家族讲席教授，Nam Ling 博士）［E21］。

另外的信息可以从 http://www. itu. int/en/ITU-T/studygroups/com16/video/Pages/jctvc. aspx 获得。

JCT-VC 在 2010 年发出联合征集提案的请求［Q20］。

- 提交了 27 份完整的提案（有些是多个组织联名的）；

- 每份提案都是一个重要的资料包——许多编码视频、大量的文档、提交的大量性能指标、有时候还有软件等；
- 大量的主观测试（3 个测试实验室、4200 个评估视频剪辑、850 个人类主观个体、300，000 个分数）；
- 建议视频的质量与 AVC（ITU-T Rec. H.264 或 ISO/IEC 14496-10）锚固编码结果相比较；
- 发布的测试报告 JCTVC-A204|/N11775；
- 在某些情况下，以 AVC（H.264）一半的比特率提供可比拟的质量；
- 源视频序列按视频分辨率从四分之一 WVGA（416×240）到大小为 2560×1600 的视频分成五类，后者裁切自 4k×2k 的以 8bpp 渐进扫描的 YCbCr 格式超高清（UHD：ultra HD）视频；
- 在“随机访问”（1s）和“低时延”（没有帧的重排序）条件下都进行了测试（表 5.1）。

图 5.2 和图 5.3 给出了在所有测试视频上的平均结果；其中，第一幅图（见图 5.2）表示在随机访问限制条件下的平均结果，第二幅图（见图 5.3）表示在低时延限制条件下的平均结果。

这些结果基于 11 个等级，0 表示最坏而 10 表示最好的个人质量衡量。图中除了每个平均意见得分（MOS）数据点之外，还有 95% 的置信区间。

测试之后所执行的更为详细的分析表明，在相当数量的情况下性能最好的建议在约一半锚固比特率下表现出类似与 AVC 锚固类似的质量［E23，E59，E97］。

2010 年四月 15-23 日，在德国德累斯顿召开首届 JCT-VC 会议上进行了对于所提交方法的技术评估。评估表明，所有的提交算法都基于传统的混合编码方法，综合了视频帧间的运动补偿预测、帧内预测、环路滤波的闭环操作、空间残差信号的 2D 变换、和先进的自适应熵编码组合。

表 5.1　用于 CfP 中的测试类和比特率（限制）

类　别	比特率 1	比特率 2	比特率 3	比特率 4	比特率 5（Mbit/s）
A：2560×1600p30	2.5Mbit/s	3.5Mbit/s	5Mbit/s	8Mbit/s	14
B1：1080p24	1Mbit/s	1.6Mbit/s	2.5Mbit/s	4Mbit/s	6
B2：1080p50-60	2Mbit/s	3Mbit/s	4.5Mbit/s	7Mbit/s	10
C：WVGAp30-60	384kbit/s	512kbit/s	768kbit/s	1.2Mbit/s	2
D：WQVGAp30-60	256kbit/s	384kbit/s	512kbit/s	850kbit/s	1.5
E：720p60	256kbit/s	384kbit/s	512kbit/s	850kbit/s	1.5

作为向协作迈进的一个初始步骤，产生了测试模型讨论稿（TMuC，Test Model under Consideration），该文档兼有从七个高性能提案中识别出的关键元素。这首个

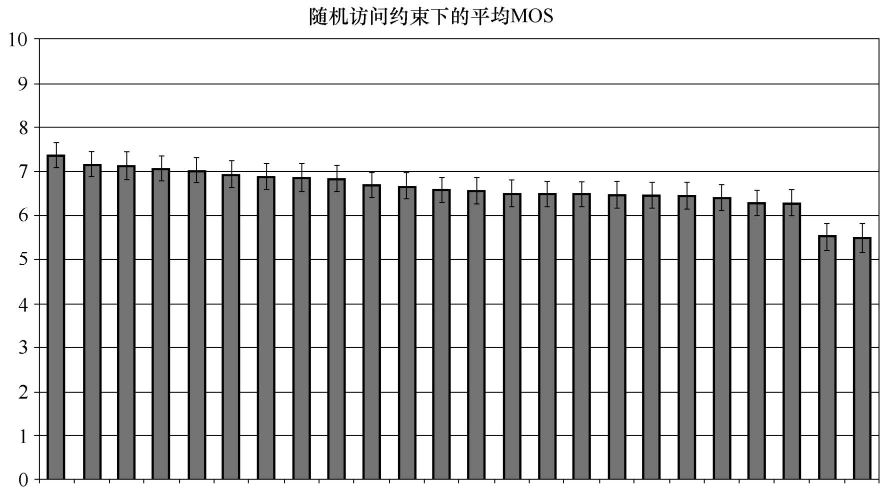


图 5.2　随机访问编码条件下所有视频类上的 MOS 结果总体平均［E5］

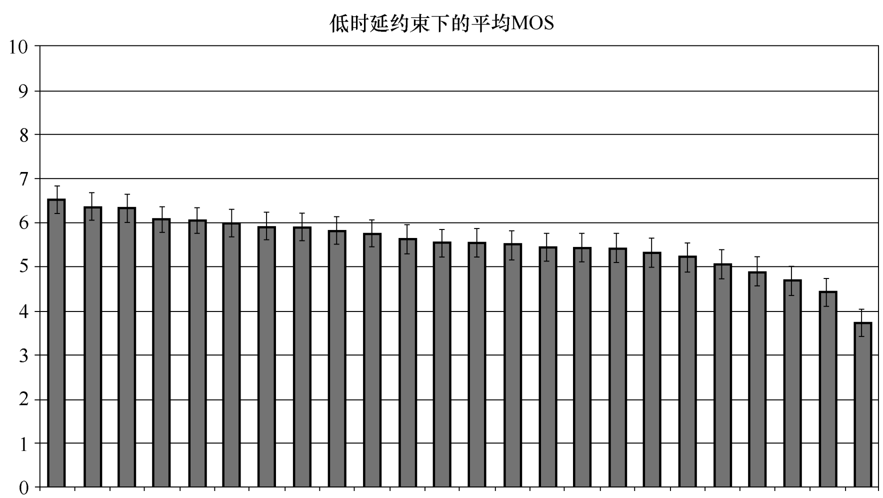


图 5.3　所有类别在低时延编码条件下的 MOS 总体平均［E5］

TMuC 成为首次软件实现的基础，在它的制订之后已经开始能够进行针对所含编码工具和附加工具更加严格的评估，这些工具在首届 JCT-VC 会议计划的“工具试验”（TE，Too Experiments）过程里要进行研究。

由于引入了更大的带有灵活亚分割机制的块结构，高分辨率视频更高压缩性能的最有利的元素之一到来了。为此，TMuC 定义了编码单元（CU，Coding Unit），它定义了将一帧图像亚分割为相同（通常）或可变尺寸的矩形区域。编码单元取替了宏块结构（H.264）且包括一个或若干预测单元（PU，Prediction Unit）和变换单元（TU，Transform Unit）。所有这些元素基本的分裂几何结构由一个类似与

四叉树分割结构的方案进行编码。在 PU 级别上，可以选择帧内编码或者帧间编码。

Kim 等的论文“Block partitioning structure in the HEVC standard”［E91］，解释了块分裂结构的技术细节并给出编码效率和复杂度的分析结果。

- 帧内预测从已经解码的相邻 PU 样值进行，其中的不同模式包括直流（平均）、水平、垂直、或多达 28 个方向（取决于块大小）的角度之一、平面（幅度表面）预测、和双线性预测。模式指示从相邻 PU 的模式中获得（见图 5.4）。

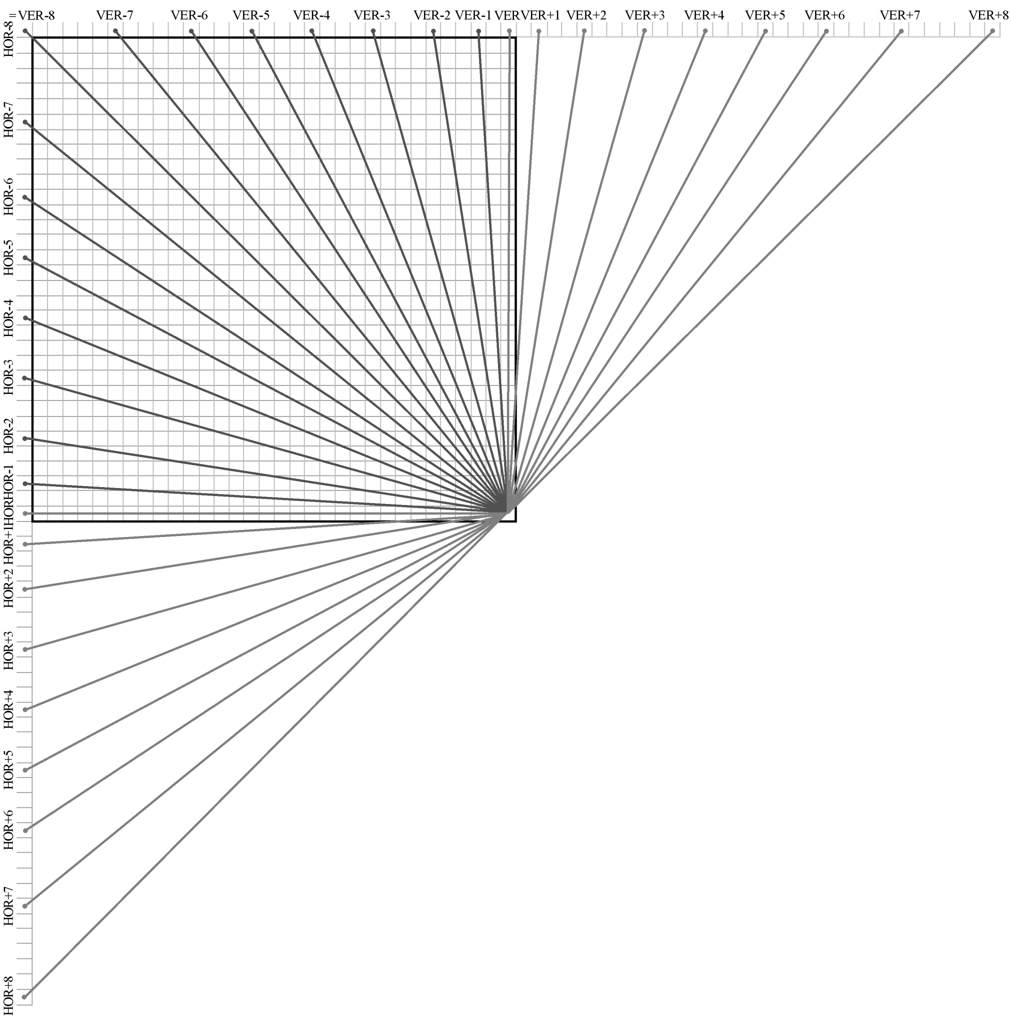


图 5.4　HM 1.0 统一帧内预测中的可用预测方向

- 帧间预测从存储在参考帧里已经解码的帧图像的区域进行。这就允许在多幅参考图像中进行选择、以及来自两幅参考帧或者同一参考帧内的两个位置上的双

向预测（包括加权平均）。在此背景下，就运动矢量（四分之一精度）的用法而言，相邻 PU 的合并是可能的，非矩形亚分裂也是有可能的。为了高效的编码，定义了类似于 H.264/AVC（第 4 章）的跳跃和直接模式，而且通过各种方法诸如中值计算或叫作运动矢量竞争的新方案来获得相邻 PU 的运动矢量。

在 TU 级别（通常不会比 PU 大）上，使用了在概念上类似与 DCT 的整数空域变换，块大小从 4×4 到 64×64 可选。对于帧内方向模式，对于通常在残差预测中呈现出方向结构的方向帧内模式，使用了专门的模式相关的方向变换（MDDT，Mode-Dependent Directional Transform）用于 4×4 和 8×8 大小的块。此外，对于大于 8×8 的块的情况，可以使用旋转变换（见 P.5.13）。变换系数值的缩放、量化和扫描以类似于 AVC 的方式进行。

在 CU 级别，有可能在一个自适应环路滤波器（ALF，Adaptive Loop Filter）上进行转换，在将帧复制到参考帧缓冲之前 ALF 用在预测环路中。这是一种 FIR 滤波器，设计的目的是最小化相对于原始帧的失真（例如，在最小二乘或维纳滤波器优化的意义下）。滤波器系数在分片级别上进行编码。此外，去块效应滤波器（类似于在 H.264/AVC 中设计的去块效应滤波器）也工作在预测环路内。在应用这两个滤波器之后，解码器的显示输出写入解码帧缓冲器。请注意，在 HEVC 标准中 ALF 被弃之不用了［E23，E59，E97］。在更新版本中，环路中的滤波包括去块效应和样值自适应偏移（SAO，Sample Adaptive Offset）滤波器（见图 5.4）。关于 HEVC 标准中的 SAO 参见［E85，E109］。

TMuC 定义了两个上下文自适应的熵编码方案，一个用于较低复杂度模式的工作中，另一个用在较高复杂度模式下。

TMuC 的实现软件已经被开发出来了。基于此，JCT-VC 正在进行有关 TMuC 包含的编码工具以及除此之外提出的其他工具性能的研究。基于这样工具试验（TE，Tool Experiments）的结果，该小组将定义一个称为测试模型（TM，Test Model）的更加有效的设计作为 HEVC 标准化过程中下一个有重大意义的步骤。有关目前 TMuC 诸元的逐工具评价的专门实验已被列入计划；此外，其他能在帧内帧间预测、变换、熵编码和运动矢量编码领域给出压缩能力或复杂度降减方面给出额外益处的工具评价也在此列。各种各样的特设团队（AHG，ad hoc groups）建立起来对诸如复杂度分析这样的问题开展额外的研究，如下所列：

建立的特设协调小组

- JCT-VC 项目管理；
- 考虑编辑中的测试模型；
- 软件开发与 TMuC 软件技术评估；
- 帧内预测；
- 其他被选的变换；
- 运动矢量精度；

- 环路滤波器；
- 大块结构；
- 并行熵编码。

总之，运动补偿变换预测混合的编码框架、自适应帧内/帧间编码、自适应方向帧内预测、多种块尺寸运动估计、SAO 滤波器［E85，E109］、环路去块效应滤波器［E71］、熵编码（CABAC，见［E65，E66］）、多帧运动补偿加权预测、从 4×4 到 32×32［E72］的整数变换、H. 264/AVC（第 4 章）引入的帧内编码直流系数哈达玛变换编码、以及如下所列的各种其他工具，综合起来与 H. 264/AVC 相比已经在编码效率、降低比特率、更高的 PSNR 方面表现出进一步的增益。这样，HEVC 在若干不同的应用领域中很有希望和潜力，而且甚至可能赶上 H. 264/AVC。

1）在帧内预测中（见图 5. 4）不同的 PU 有各种方向预测模式（多达 34 种）［E49、E102］；

2）对于帧内编码，除了传统的水平/垂直扫描之外还有模式相关的方向变换（MDDT，mode dependent directional transform）［E6、E49、E15］；

3）面向大于（8×8）的块的旋转变换（见 P. 5. 13）；

4）高达 32×32 的大尺寸变换［E6，E72］；

5）环内去块效应滤波器［E71］和 SAO 滤波器［E85，E109］；

6）面向运动估计/运动补偿的大尺寸块；

7）非矩形运动分裂［E59］。

注意，其中第 2 项和第 3 项已被弃而不用，因为它们对于编码效率贡献非常小却以大幅增加复杂度为代价。其他被提出的方向变换，像方向 DCT［E110］和模式相关 DCT/DST［E111］，也因为同样的原因被放弃了。

附笔：JCT-VC 的介绍基于 Sullivan 和 Ohm 发表的论文 “Recent developments in standardization of high efficiency video coding（HEVC）”，数字图像处理的应用 XXXIII，SPIE 会议录，第 7798 卷，页码：77980 V-1 到 7980 V-7，2010.

对于 HEVC 的最新进展，读者可参考一篇优秀的评论论文：Sullivan et al.，“Overview of high efficiency video coding（HEVC）standard”，IEEE Trans. CSVT，vol. 22，pp. 1669-1684，Dec. 2012［E50］. 还有关于 HEVC 主题报告［E97］、HEVC 指南［E23］、HEVC 文本规格草案 8［E58］。

对于帧内模式，从 DST 推导得出的另一种变换只用于 4×4 亮度块。对于其他情况使用整数 DCT。

5. 3　HEVC 测试模型中编码工具的分析，HM 1. 0：帧内预测

在 HM 1. 0 中，对于不同的 PU，统一帧内预测提供多达 34 种方向预测模式。

对于大小为 4 ×4，8 ×8，16 ×16，32 ×32 和 64 ×64 的 PU，分别有 17，34，34，34，和 5 种可用的预测模式。统一帧内预测中的预测方向的角度为 ± [0，2，5，9，13，17，21，26，32] /32。在垂直预测的情况下，角度由 PU 的底行和 PU 上的参考行给出；或者在水平预测的情况下，角度由 PU 最右列的位移和 PU 左侧的参考列给出。图 5.4 示出了一个 32 ×32 块预测方向的例子。对于所有的块大小，像素的重建以 1/32 的像素精度使用顶上或左侧的参考样值进行线性内插，而不是对于不同尺寸具有不同的精度。

更多有关 HM 1.0 统一帧内预测的细节可以访问 http://www.h265.net/2010/12/analysis-of-coding-tools-in-hevc-test-model-hm-intra-prediction.html。

HEVC 的工作草案（WD，Working Draft）已经经历几次更新/修订，而且最终草案国际标准（FDIS，Final Draft International Standard）已经在 2013 年 1 月出现了。总的目的是，在适度增加编码器/解码器复杂度的情况下，与 H.264/AVC（第 4 章）相比降低比特率，显著提高 PSNR。

5.4 HEVC 编码器

IEEE Trans. CSVT vol. 22，Dec. 2012 是关于下一代视频编码中新兴的研究和标准的专刊 [E43]。这期专刊提供了 HEVC 相关技术、实现和系统的最新进展，重点是进一步的研究。因为 HEVC 正在发展之中，本章包括若干有关 HEVC 的专题，附带适当的参考文献；以及关于 KTA 和 HEVC 软件的信息 [E95]。这些专题很有希望能提供那些提出的工具和技术的更深一步的洞察力，而且提供了针对它们的修改的专题讨论会，这些修改导致了 HEVC 编码器/解码器的进一步改善。图 5.5 描述了 HEVC 编码器框图 [E59]。这没有给出代表各种模式（帧内/帧间、CU/PU/TU 尺寸、帧内角度预测方向/模式、运动矢量预测、变换系数的缩放和量化，以及其他模式在解码器框图中给出——见图 5.6）的编码比特流。

对于方向帧内模式，与 DST 相关的另一种变换用于 4 ×4 亮度预测残差。对于所有其他情况使用整数 DCT（帧内/帧间色度，和帧内亮度）。2012 年 7 月，这种改变在斯德哥尔摩被采纳。在熵编码方面，只采用上下文自适应二进制算术编码（CABAC，Context Adaptive Binary Arithmetic Coding）[E65，E66]，而不像 H.264/AVC（第 4 章）中的两种（CAVLC 和 CABAC）。[E58] 提供了上下文建模的细节、自适应系数扫描和系数编码。模式相关的方向变换 [E3，E15] 没有被采用。除了 4 ×4 帧内亮度外的所有情况都只使用 INTDCT（可分离的 2 维）。

尽管 HEVC 遵循传统（也经过检验的）的基于块的运动补偿预测，然后是变换编码、量化、变长编码（也是自适应帧内模式），由于它采取了灵活的四叉树编码块分裂结构而具有明显的不同。除了去块效应滤波器 [E71] 之外，大尺寸的块

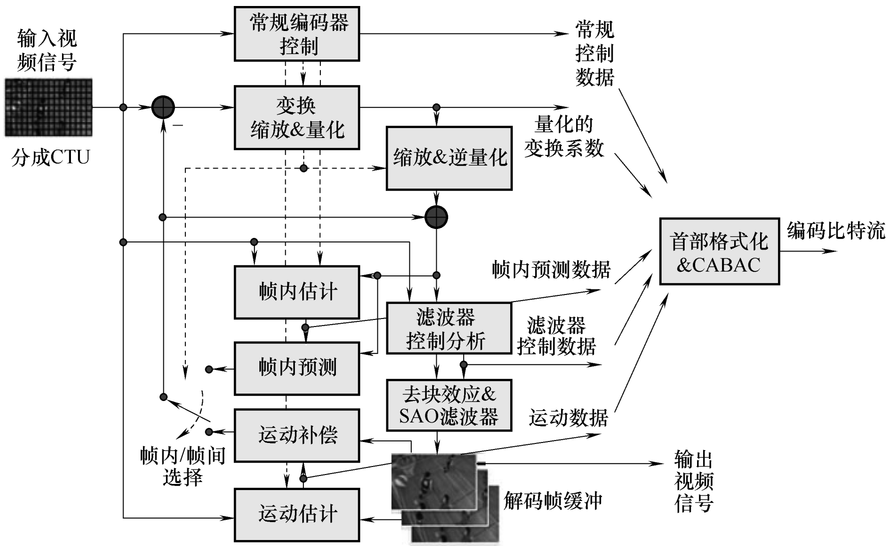


图 5.5　HEVC 编码器框图［E59］（2012 IEEE）

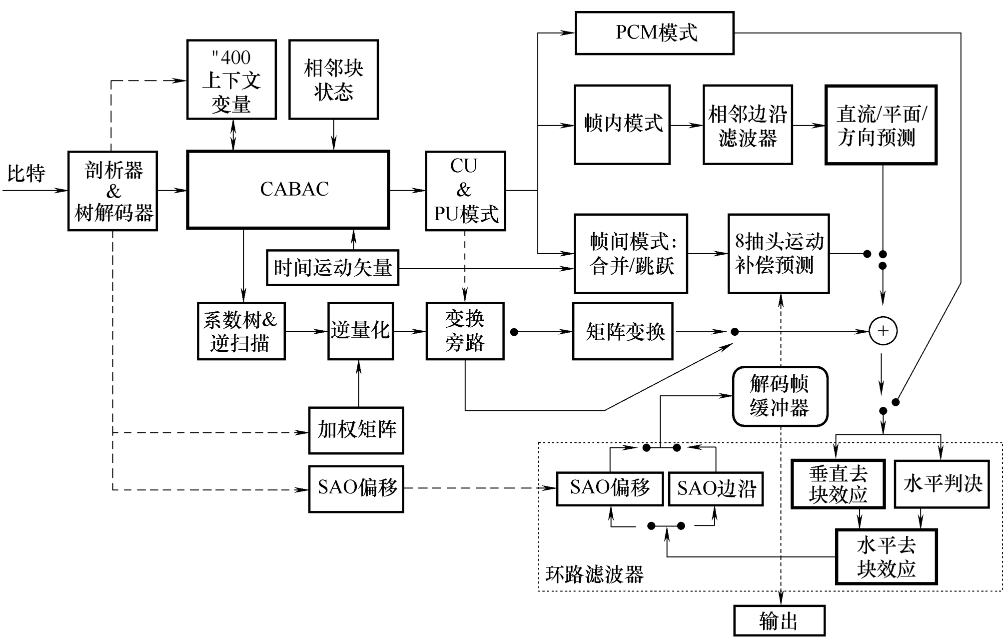


图 5.6　HEVC 解码器框图［E22］

变换（高达 32×32）、先进的运动预测、样值自适应偏移（SAO，Sample Adaptive Offset）［E85，E109］扩大了递归树的编码结构。大的多尺寸递归结构归类为编码单元（CU，Coding Unit）、预测单元（PU，Prediction Unit）和变换单元（图 5.7）

［E42］。参见［E72］获得有关 HEVC 中变换系数编码的细节。

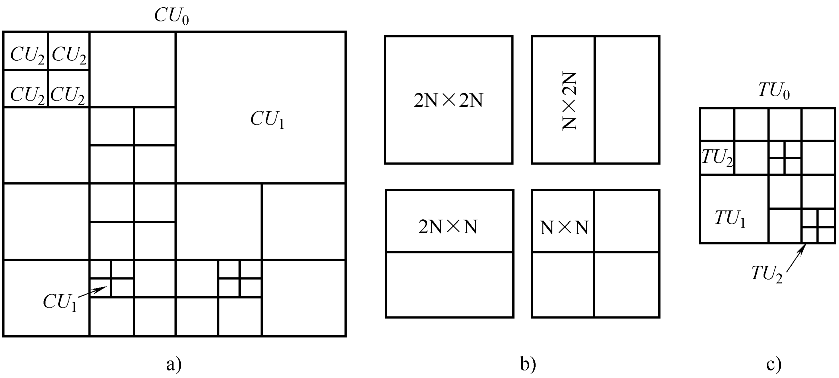


图 5.7 HEVC 的递归块结构（其中 k 表示 CUk 和 TUk 的深度）［E42］
a）CU 结构 b）PU 结构 c）TU 结构

通过引入帧内自适应角度方向预测、模式相关上下文样值平滑、自适应运动参数预测、环路滤波（去块效应滤波器和 SAO），进一步增加了 HEVC 编码器复杂度。这些和其他工具贡献了超越 H.264 编码效率改善的 50%，代价是编码器复杂度的大幅提高。然而，解码器复杂度类似于 H.264/AVC［E23，E59，E103］。有人建议了降低编码器帧内预测复杂度的几种技术（参见［E42］和最后引用的各种参考文献）。并且参见［E106］。Zhang 和 Ma［E42］也在探索降低帧间预测的复杂度。他们建议这两个方面（帧内和帧间预测模式）可被组合起来降低 HEVC 编码器的整体复杂度（参见专题 P.5.16 到 P.5.19）。这是一片研究的肥沃土壤。Main 和 high efficiency 10（HE 10）包含的工具总结在表 5.2 中。这些工具的细节在测试模型编码器描述中［E57］有详细说明。另外参见评论文章［E59，E97，E105］。论文［E59］指出"为了帮助工业界学会如何使用标准，标准化工作不仅包括文本性的说明书文档进展，也要包括参考软件源码（编码器/解码器）"［E95］。这个软件可作为一个研究工具和产品的基础。该论文也指出"标准的测试数据集也在形成，用于测试与标准的一致性"。

5.4.1 帧内预测

图 5.8 所示为对应于图 5.4 描述的 VER 和 HOR 的 33 个帧内预测方向的角度［E57，E59，E123］。图 5.9 示出了 33 个帧内预测模式方向。帧内预测模式方向和角度直接的映射如表 5.3 所示［E123］。也要参见［E102］。这些帧内预测模式对于 HEVC 的性能改善具有显著的贡献。对于所有帧内预测的情况，使用方向预测模式的统计分析表明，除平面（模式 0）和直流（模式 1）外，水平（模式 10）和垂直（模式 26）位于前列［E102］。［E102］的作者们通过制订一个角度表已经表明，对于含有大量丰富纹理的视频序列能提高编码增益。每个帧内编码的 PU 对

亮度会有一个帧内预测模式，对色度分量会有另一个模式。一个 PU 内所有的 TU 对每个分量会使用相同的相关模式。然后，编码器从 35（33 加上平面和直流）个方向中选择最好的亮度帧内预测模式。由于预测方向数目的增加（与 H. 264/AVC——第 4 章）相比，HEVC 考虑了三个最可能的模式（MPM，Most Probable Mode），相比之下 H. 264/AVC 只考虑一种 MPM。对于帧内编码 PU 的色度分量，编码器从平面、直流、水平、垂直和亮度帧内预测模式的直接复制这 5 种模式中进行选择最好的色度预测模式，[E57，E123] 给出了色度帧内预测方向和模式#的映射细节。

[E59、E97、E105] 提供了 HEVC 编码器相关的分片、编码单元、预测单元、变换单元、帧间预测模式、特殊编码模式、运动矢量估计/预测、分数像素精度的运动矢量内插滤波器、加权预测、变换的大小（4×4，8×8，16×16，32×32）[E72]、变换系数扫描——图 5. 10（详见 [E72]）、缩放/量化、环路滤波器（去块效应滤波器 [E71] 和 SAO [E85，E109]）、熵编码（CABAC [E65，E66]）和其他各种各样功能的详细描述。流行的 Z 形扫描没有被采用。

表 5. 2　HM9 配置中的工具结构

Main	High efficiency 10（HE10）
高级别结构：	
对于帧率时间嵌套和随机访问的高级别支持	
清楚随机访问支持	
矩形分片结构的扫描	
面向并行性的波阵面结构处理依赖性	
等于码树单元的空间粒度分片	
具有独立和非独立分片部分的分片	
编码单元、预测单元、和变换单元：	
编码单元四叉树结构方形编码单元块大小 $2N\times2N$，$N=4$，8，16，32（即大小高达 64×64 的亮度样值）	
预测单元（对于大小为 $2N\times2N$ 的编码单元：）对帧间编码，$2N\times2N$，$2N\times N$，$N\times2N$，对 $N>4$ 时 $2N\times(N/2+3N/2)$ 和 $(N/2+3N/2)\times2N$；对帧内编码，只有 $2N\times2N$ 和 $N=4$ 时也有 $N\times N$	
编码单元内的变换单元树结构（最多 3 级）	
4×4 到 32×32 样值的变换块尺寸（总是方形的）	
信号的空域变换与PCM 表示：	
类似 DCT 的整数块变换；对于帧内编码也可用一种基于 DST 的整数块变换（仅对 4×4 的亮度块）	
对帧间编码变换可能跨越预测单元的边界；对帧内编码则不会	
对 4×4 的变换单元允许跳跃变换	
带有最坏情况比特使用限制的 PCM 编码	

（续）

Main	High efficiency 10（HE10）
内部编码帧预测：	
帧内编码角度预测（35 种模式包括直流和平面模式）	
帧内编码平面预测	
帧间编码帧预测：	
亮度分量运动补偿内插：1/4 样值精度，对 1/2 样值精度的 8×8 可分离 6bit 抽头值，对于 1/4 精度的 7×7可分离 6bit 抽头值	
色度分量运动补偿内插：1/8 样值精度，4×4 可分离 6bit 抽头值	
带有运动矢量“竞争”和“合并”的先进运动矢量预测	
熵编码：	
上下文自适应二进制算术熵编码（CABAC）	
率失真优化的量化（RDOQ）	
帧存储与输出精度：	
8bit/样值的存储和输出	10bit/样值的存储和输出
环路滤波：	
去块效应滤波器	
样值自适应偏移滤波器（SAO）	

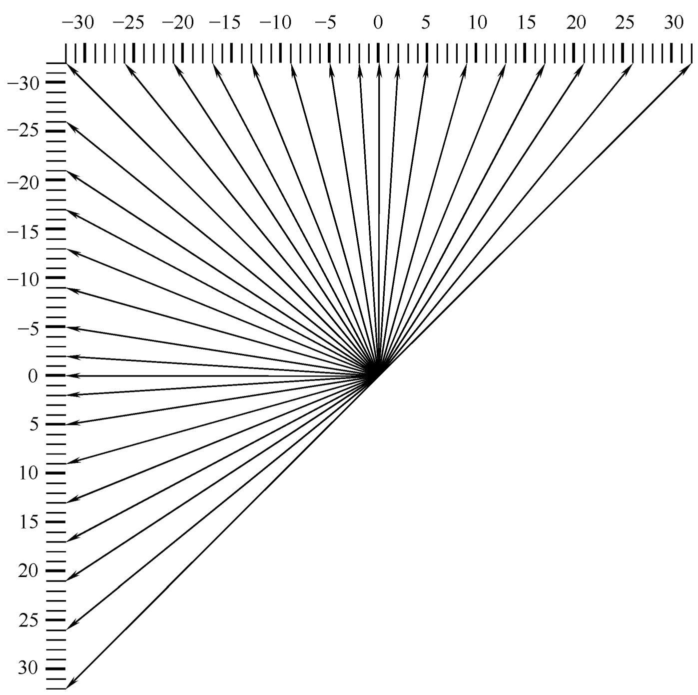


图 5.8　帧内预测角度预测［E57，E123］

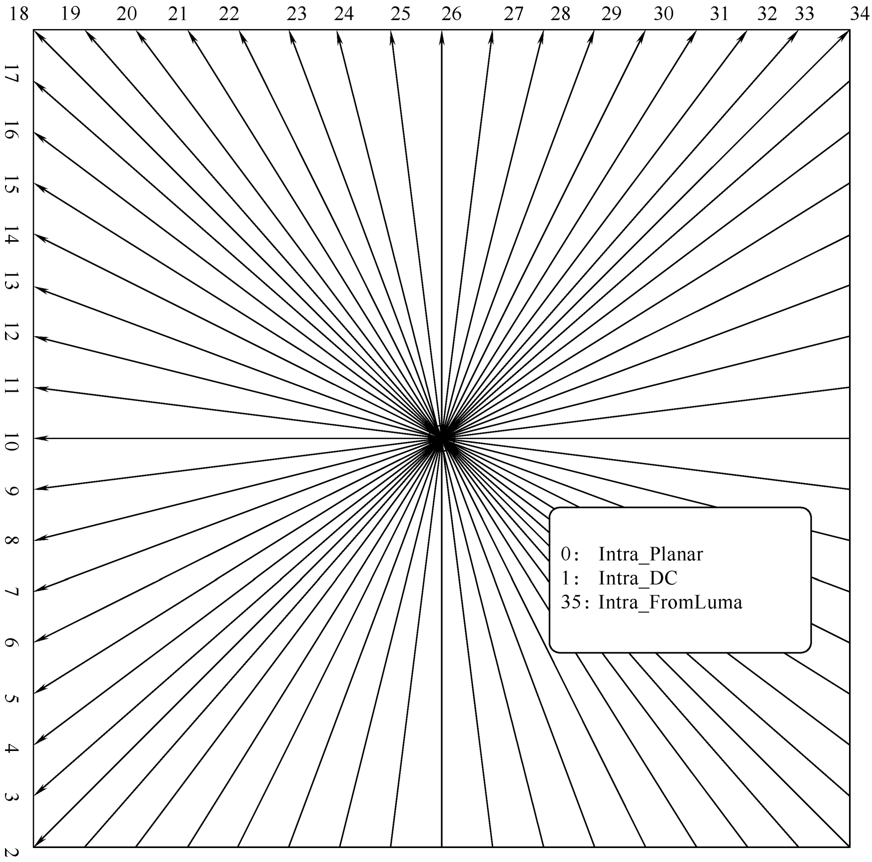


图 5.9　帧内预测模式的方向［E57，E123］，另见［E102］

表 5.3　帧内预测模式方向（见图 5.9）**和帧内预测角度**（见图 5.8）**直接的映射**［**E123**］

帧内预测模式	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
帧内预测角度	–	32	26	21	17	13	9	5	2	0	–2	–5	–9	–13	–17	–21	–26
帧内预测模式	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34
帧内预测角度	–32	–26	–21	–17	–13	–9	–5	–2	0	2	5	9	13	17	21	26	32

5.4.2　变换系数扫描

HEVC 对于 8×8 变换块采用的三种变换系数扫描方法，对角、水平和垂直如图 5.10 所示［E72］。4×4 变换块的扫描是对角的。在帧内编码情况下对于 4×4 和 8×8 变换块可能会使用水平和垂直扫描。

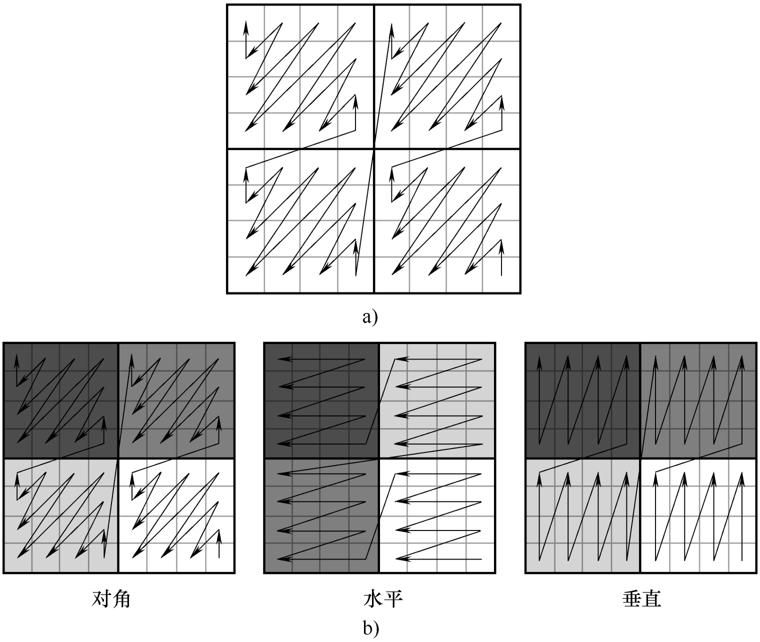


图 5.10 a）8×8 变换块的对角扫描模式：4×4 变换块的对角扫描用在大块的每个 4×4 子块内 b）8×8 变换块的系数组［E72］（2012 IEEE）

5.4.3 亮度与色度的分数像素内插

亮度分量内插的整数（$A_{i,j}$）和分数像素位置（下标字母）如图 5.11 所示［E59，E107］。对于一般的内插参见［E69］。

不像 H.264 采用的两阶段内插过程，HEVC 对于 1/2 像素使用可分离的 8 抽头滤波器，对于 1/4 像素采用 7 抽头滤波器（表 5.4）［E59，E69，E107］。类似地，色度分量分数（1/8 精度）像素内插的 4 抽头滤波器系数如表 5.5 所列。Lv 等［E107］已经进行过针对 HEVC 和 H.264/AVC 中的分数像素内插滤波器的性能比较研究，并得出结论：HEVC 的滤波器比 H.264/AVC（第 4 章）的滤波器提高了超过 10% 的 BD 率［E79，E80，E94］，代价是增加了实现的复杂度。

表 5.4 亮度的分数样值内插滤波器系数［E59］

下　标	−3	−2	−1	0	1	2	3	4
hfilter［i］	−1	4	−11	40	40	−11	4	1
qfilter［i］	−1	4	−10	58	17	−5	1	

（IEEE 2012）

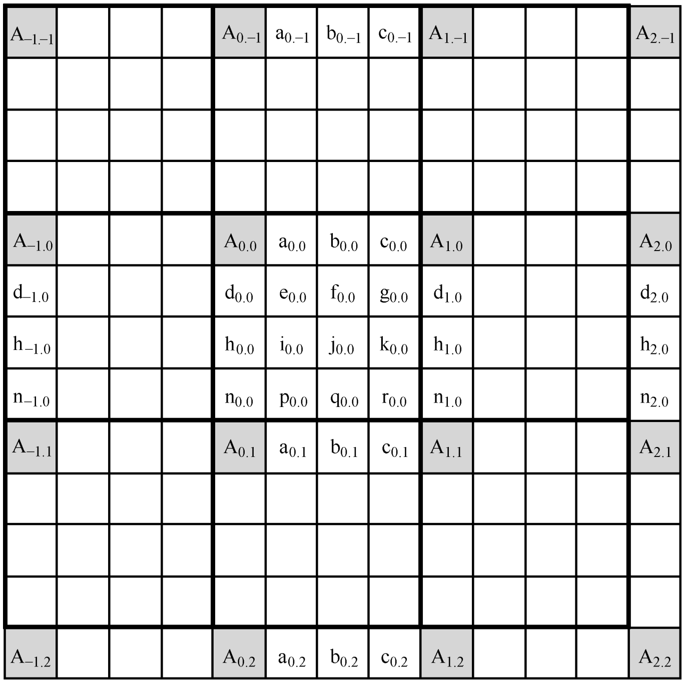


图 5.11　亮度内插的整数和分数位置（2012 IEEE）

表 5.5　色度的分数样值内插滤波器系数［E59］

下　标	−1	0	1	2
filter1［i］	−2	58	10	−2
filter2［i］	−4	54	16	−2
filter3［i］	−6	46	28	−4
filter4［i］	−4	36	36	−4

（IEEE 2012）

5.4.4　HM1 与 HEVC 草案 9 编码工具的比较

HEVC 测试模型第 1 版（HM1）和草案 9［E57］的编码工具总结在表 5.6 中［E59］。Sullivan 等人关于 HEVC 的综述论文［E59］是一个卓越的文献资源，它不仅澄清了所有固有的功能，而且还说明了导致这个相当有效的标准的历史和标准化过程。从长远来看，HEVC（包括附加内容/扩展/档次）已经具有潜力/前景赶上所有以前包括 H.264/AVC（第 4 章）的标准。

5.5　扩展到 HEVC

与 H.264/AVC 一样，HEVC 的附加/扩展内容包括 4:2:2 和 4:4:4 格式、更高

表 5.6 HM1 和 HEVC 中高效配置的编码工具的总结［E59］

功　　能	HM1 高效率	HEVC（草案 9）
CTU 结构	从 8×8 倒 64×64 的树结构	从 8×8 倒 64×64 的树结构
PU 结构	方形对称的	方形、对称和非对称（方形只对帧内编码）
TU 结构	方形 TU 的树结构	方形 TU 的树结构
核变换	从 4 到 32 点的整数变换（完全因数分解）	从 4 到 32 点的整数变换（部分可分解）
其他的变换	不适用	对于 4×4 的整数 DST 类型
帧内预测	33 种角度模式加直流模式	33 种角度模式加平面和直流模式
亮度内插	12 抽头可分离	8 抽头/7 抽头可分离
色度内插	双线性	4 抽头可分离
运动矢量预测	AMVP	AMVP
运动补偿合并	空域 CU 合并	PU 合并
熵编码	CABAC	CABAC
去块效应滤波器	非并行的	并行的
样值自适应偏移	不适用	启用
自适应环路滤波器	多种形状	不适用
并行处理的专用工具	分片	分片、小片、波阵面和相关的分片部分

的位深（10 和也许 12）、可扩展视频编码（SVC，Scalable Video Coding）［E67，E68，E73，E74，E117］、3D/立体/多视编码（这些内容的一部分正被探索中）。参见［E21，E23］。与 SVC 有关的几个议案已于 2012 年 10 月在上海召开的 HEVC（ITU-T/ISO，IEC）会议上提交了。读者可以参考海报论文［E25］MA. P2 “High efficiency video coding”，IEEE ICIP 2012，佛罗里达州奥兰多市，9-10 月，2012 和下一代视频编码的新兴研究与标准专刊，IEEE Trans. CSVT，22 卷，2012 年 12 月［E43］。这期专刊的几篇文章在此作为参考文献加以引用［E59-61，E65，E67，E68，E71，E72，E74-E76，E81-E83，E85-E91，E118］。这些论文不仅涵盖了 HEVC 的各个方面，而且还涵盖了降低实现复杂度的途径和该标准采纳各种工具和方法背后的原因。其中的部分论文也提供了最初考虑某些工具的理由，诸如自适应环路滤波器、MDDT、ROT 等，以及后来标准去除这些工具的理由。IEEE 期刊 IEEE Journal of selected topics insignal processing 也进行“视频编码：HEVC 及其他”的专题征文，2013 年 6 月 25 日截稿［E78］。

5.6　档次和级别

目前，在国际标准终稿（FDIS，Final Draft International Standard）（2013 年 1 月）列出了三个档次（主档次、主 10 档次和主静止图像档次——只用于帧内编码帧）［E123］。ITU-T 第 16 研究组已经同意这个第一阶段的认可案稿正式叫作 H.265 建议或 ISO/IEC 23008-2。主档次限于 YCbCr 4∶2∶0 格式、8 比特位深、渐进扫描（非隔行的）、空间分辨率范围从 QCIF（176×144）到 7640×4320（叫作 8k×4k）。

图 5.12［G17］列出了从标清（NTSC）到高品质电视/超高清视频的空间分辨率范围。在主档次的第一版中包括了 13 个级别（表 5.7）［E59，E97］。

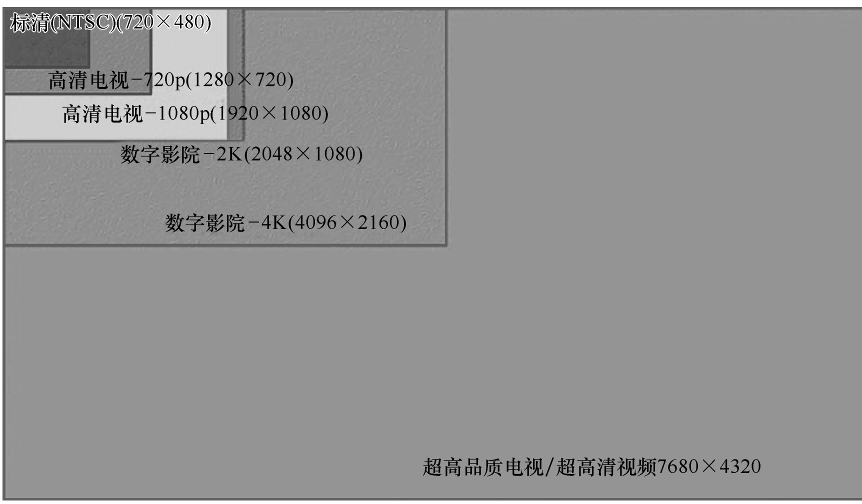


图 5.12　未来的视觉应用与需求

表 5.7　HEVC 主档次的级别限制［E59］

级别	最大亮度帧尺寸（样值数）	最大亮度样值速率（样值/s）	主层级最大比特率（1000bit/s）	高层级最大比特率（1000bit/s）	最小压缩比
1	36 864	552 960	128	—	2
2	122 880	3 686 400	1 500	—	2
2.1	245 760	7 372 800	3 000	—	2
3	552 960	16 588 800	6 000	—	2
3.1	983 040	33 177 600	10 000	—	2
4	2 228 224	66 846 720	12 000	30 000	4

（续）

级别	最大亮度帧尺寸（样值数）	最大亮度样值速率（样值/s）	主层级最大比特率（1000bit/s）	高层级最大比特率（1000bit/s）	最小压缩比
4. 1	2 228 224	133 693 440	20 000	50 000	4
5	8 912 896	267 386 880	25 000	100 000	6
5. 1	8 912 896	534 773 760	40 000	160 000	8
5. 2	8 912 896	1 069 547 520	60 000	240 000	8
6	33 423 360	1 069 547 520	60 000	240 000	8
6. 1	33 423 360	2 005 401 600	120 000	480 000	8
6. 2	33 423 360	4 010 803 200	240 000	800 000	6

5. 7 HEVC 编码器的性能和计算复杂度

对于 HEVC 编码器的编码效率性能对计算复杂度的关系，Corra 等人［E82］已经进行了一次彻底而详细的研究。这项研究重点集中在找出最为影响这两个关键参数（效率和复杂度）的工具。一项非常宝贵的成果就是效率和复杂度直接的均衡点，这对于实现复杂度受限的编码器而言是非常有用的。

此外，实现与高复杂度配置可比拟的编码效率的低复杂度编码器的进展对于工业界也是一个宝贵资源。通过使用哈达玛运动估计、非对称运动分割和环路滤波器而不是要求很多计算量的工具如运动估计，可以同时实现低复杂度和高效率。他们的分析包括的三个编码工具（非方形变换、自适应环路滤波器和 LM 色度）后来从标准草案中去掉了［E58］。

5. 8 HEVC 的系统层面集成

Schierl 等人描述了 HEVC 的系统层面集成［E81］，讨论了将 HEVC 集成到端到端多媒体系统、格式和协议（RTP、MPEG-2 TS、ISO 文件格式和 HTTP-DASH 上的动态自适应流），也提到了 HEVC 中的抗误码工具。H. 264/AVC 的许多抗误码工具诸如 FMO、ASO、冗余分片、数据分裂以及 SP/SI 帧（第 4 章）由于它们的利用率都被去除了。他们建议，HEVC 错误隐藏的使用在实现中应该被仔细考虑，而且是一个未来研究的课题。他们描述了不同系统功能层提供的视频传输和交付诸如广播、网络电视、网络流媒体、视频会话和存储，并以此作为结束。参见 Schierl 等：“Working draft 1. 0 for transport of HEVC over MPEG-2 Systems”，ISO/IEC SC29/WG11，MPEG99/N12466”2012 年 2 月。

5.9　HEVC 的无损编码与改进

Zhou 等［E83］已经实现了 HEVC 主档次的无损编码模式（带通变换、量化和环路滤波器——图 5.13［E83］），而且显示出了显著的性能改善（比特率下降），超越了当前无损方法诸如 JPEG2000、JPEG-LS、7-Zip 和 WinRAR（见附录 F）。他们通过引入基于样值的帧内角度预测（SAP，Sample Based Intra Angular Prediction）进一步提高了编码效率。

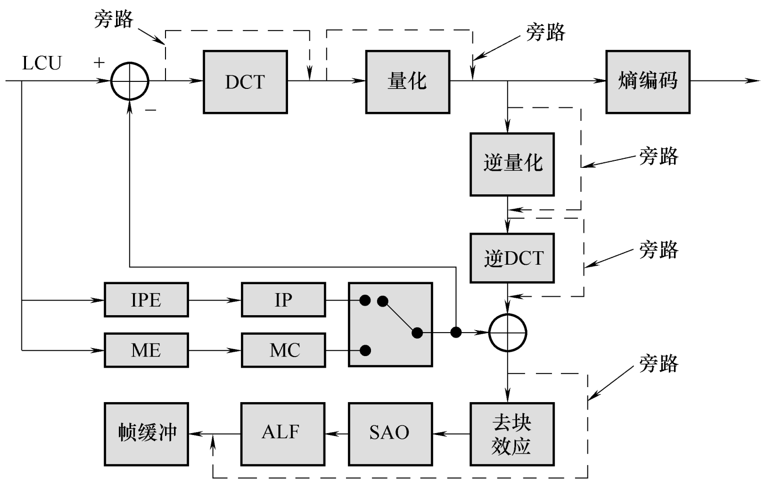


图 5.13　HEVC 无损编码模式的编码器框图［E83］（对比图 5.5）（2012 IEEE）

利用一个在 HEVC 标准发展过程中推荐的视频序列集，Cai 等［E112］也进行了 HEVC 帧内模式、H.264 高档次 4∶4∶4 的帧内模式，MJPEG2000 和 JPEG-LS 的无损编码。他们的结论是 HEVC 的性能比得上 H.264/AVC，而且与 JPEG-LS 和 MJPEG 2000 相比拟。针对帧内有损编码的类似测试表明，HEVC high 10、H.264/AVC 高档次 4∶4∶4 和 HEVC 主档次也有类似性能。然而，在低比特率环境下 MJPEG 2000 性能超越了前面三个，尽管这种优势随着比特率的提高逐渐被补偿并且最终被前三个超越。［E112］中列出了几篇其他有关这些方法和其他标准性能比较的有趣论文。

Horowitz 等［E64］进行了 HEVC MP（参考软件 HM 7.1）和 H.264/AVC HP 参考编码器（JM 18.3）在低时延应用时非正式的主观质量（双盲方式）对比。表 5.8 描述了用 HM 和 JM 编码器编码的测试序列对比结果。他们得出结论，在 73.6% 的测试当中，在大约一半比特率上，对于支持 HEVC 的观察者而言，与 H.264/AVC 相比 HEVC 一般会产生更好的主观质量。

为了强化这些结果，将 x264（产品质量级的 H.264/AVC）与 eBrisk Video

(产品质量级的 HEVC 实现方案)。参见表 5.9 的比较结果。在 62.4% 的测试当中,观察者们更喜欢约为 x264 一半比特率的 eBrisk 编码的视频。这些试验证实了 HEVC 能在 H.264/AVC 一半的比特率上产生类似的主观质量。

表 5.8 用 HM 和 JM 编码器编码的视频主观观察对比结果

HM 对 JM 的主观观察结果									
序列	HM 比特率(kbit/s)	HM 量化参数	JM 比特率(kbit/s)	JM 量化参数	HM:JM 比特率(%)	支持 HM 的投票	支持 JM 的投票	% 支持 HEVC(%)	投票总数
KnstenAndSara	149	38	302	37	49	10	15	40	25
Vidyol	190	37	367	36	52	14	11	56	25
OldTownCross	408	37	879	37	46	22	3	88	25
Kimonol	632	36	1,404	35	49	21	4	84	25
Toys_and_calendar	347	37	734	38	47	25	0	100	25
				平均:	49	92	33	73.6	125

表 5.9 用 eBrisk 和 x264 编码器编码的序列主观观察对比结果

eBrisk 对 x264 的主观观察结果									
序列	eBrisk 比特率(kbit/s)	eBrisk 量化参数	x264 比特率(kbit/s)	x264 量化参数	eBrisk:x264 比特率(%)	支持 eBrisk 的投票	支持 x264 的投票	% 支持 HEVC(%)	投票总数
KnstenAndSara	332	36	657	33	51	12	13	48	25
Vidyol	363	36	773	32	47	10	15	40	25
OldTownCross	904	35	1, 716	34	53	22	3	88	25
Kimonol	1,334	35	2, 670	32	52	17	8	68	25
Toys_and_calendar	729	36	1, 553	33	47	17	8	68	25
				平均:	50	78	47	62.4	125

5.10 总结

在视频编码标准的家族当中(图 5.1),HEVC 有指望有潜力替代/补充所有的现存标准(MPEG 和包括 H.264/AVC 的 H.26x 系列)。尽管 HEVC 编码器的复杂度是 H.264/AVC 的好几倍(第 4 章),但解码器复杂度却没有超出后者。研究者们正在探索有关降低 HEVC 编码器的复杂度的问题[E61, E77, E82, E86, E87, E104, E106]。回报是几个测试已经显示,HEVC 改善了压缩效率,对于可比拟于

H. 264/AVC（第 4 章）的同样主观质量，可提供高达 50% 的比特率减少量［E59，E97］。参见表 5. 8 和 5. 9。除了解决所有目前的应用之外，设计和制订 HEVC 的目的集中在两个关键问题上：增加的视频分辨率——高达 8k × 4k——使用并行处理结构的增加。虽然提供了 HEVC 的简述，但对于细节和实现，读者可以参考 JCT-VC 文档［E53］、综述论文［E5，E59，E105］、主题报告［E21，E97］、教程辅导［E23］、专题讨论会［E98］、海报论文［E25］、专刊［E12，E43，E78］、测试模型（TM/HM）［E54］、网址/ftp 地址［E6，E51，E52］、开源软件［E47，E69，E95］、测试序列和锚定比特流［E49］。研究者们［E93，E96］也在探索 HEVC 和其他标准诸如 MPEG-2 之间的转码问题。下面列出的专题涵盖了从研究生 3 学时的课程到理学硕士和博士阶段的研究内容。设计这些专题旨在提供对于方法和技术别样的理解，而且提供了一个讨论机会，讨论使得 HEVC 编码器/解码器进一步改善的修正措施。多媒体研究组公司（Multimedia Research Group Inc.）预测，2016 年之前具有 HEVC 解码能力的用户设备超过20 亿个单元（www. mrg. com）。

5. 11 专题研究

P. 5. 1 Deng 等人［E15］给 H. 264/AVC FRExt 加入了进一步的扩展，诸如更大的运动矢量搜索范围、更大的宏块、跳过的块尺寸和一维 DDCT。他们用高分辨率（HR，high Resolution）（4096 × 2160）视频序列比较了它和运动 JPEG 2000 的性能，结果表明，在各种比特率下前者的 PSNR 有明显的提高。请用 HR 测试序列实现扩展的 H. 264/AVC（第 4 章）和运动 JPEG 2000（附录 F）并证实前者具有优越的性能。Deng et al. ，“Performance analysis，parameter selectionand extension to H. 264/AVC FRExt for high resolution video coding”，J. VCIR，vol. 22，pp. 687-760，Feb. 2011.

P. 5. 2 Karczewicz 等［E8］通过加入额外的特征诸如扩大的块尺寸（高达 64 ×64）、帧内编码的模式相关方向变换（MDDT）、色度和亮度高精度滤波、自适应系数扫描、扩大的块分裂尺寸、自适应环路滤波、大尺寸整数变换等，提出了一个优于 H. 264/AVC（第 4 章）的混合视频编解码系统。通过在不同的空间分辨率上使用几个测试序列，他们给出，这个新的编译码系统在主观质量和客观指标上都超越了传统的 H. 264/AVC 编译码系统（第 4 章）。并且这只需要在编码器和解码器的复杂度上有适当的增加。请实现这个新的编译码系统并且得出类似于这篇文章中所描述的结果。也考虑把 SSIM（附录 C）作为在所有仿真中的另一个指标。请使用最新的 H. 264/AVC 实现软件 JM。Karczewicz et al. ，“A hybrid video coder based on extended macroblock sizes，improved interpolation，and flexible motion representation”，

IEEE Trans. CSVT，vol. 20，pp. 1，698-1，708，Dec. 2010.

P. 5. 3 Ma 和 Segall［E18］开发了一个 HEVC 的低分辨率（LR，Low Resolution）解码器。这里的目的是为手持或移动设备在高分辨率比特流内提供一个低功率解码器。通过在 HEVC 的框架之内采用混合帧缓冲压缩、LR 帧内预测、级联运动补偿和环路去块效应［E71］，这个目标得以推进。请实现这个低功率 HEVC 解码器；将这些工具移植到 HEVC 参考模型中（HM9. 0）［E54］并评价其性能。Ma and Segall，"Low resolution decoding for high efficiency video coding"，IASTED SIP 2011，pp.，Dallas，TX，Dec. 2011.

P. 5. 4 Joshi 等［E6］开发了缩放的整数变换，它数值上具有稳定性，结构上具有递归性，而且是正交的。他们也把这些变换嵌入进 H. 265/JMKTA 的框架里。请专门开发 16 点的缩放变换并且用 JMKTA 软件在 H. 265 中加以实现。再开发 32 和 64 点的缩放变换。Joshi et al. "Efficient large size transforms for high-performance video coding"，Applications of Digital Image Processing XXXIII，Proc. of SPIE，vol. 7798，77980W-1 through 77980W-7，2010.

P. 5. 5 请从 MPL 网址 EE5359 访问 S. Subbarayappa 的学位论文（2012）"Implementation and Analysis of Directional Discrete Cosine Transform in Baseline Profile in H. 264"。为所有有关 4 ×4 和 8 ×8 DDCT 的方向模式获取基图像。模式 4、6、7 和 8 可以从模式 3 和 5 中获得，如图 13 ~16 所示（专题）。另外参见［E110］。用这个方法获取基图像。请访问 http://www. h265. net/2009/9/mode-dependent-directional-transform-mddt-in-jmkta. html。

P. 5. 6 请访问网址 http://www. h265. net/并去分析 HEVC 测试模式（HM 1. 0）中的编码工具——帧内预测。它描述了对于不同 PU 多达 34 种方向预测模式可被用在 H. 265 的帧内预测中。请用 TmuCHEVC 软件［E95］实现 HM 1. 0 里面的这些模式并评价 H. 265 的性能。（HM：HEVC 测试）

P. 5. 7 请考虑各种比特率下的各种测试序列，用 TMuC HEVC 软件［E95］来实现 HM 1. 0。用 SSIM（附录 C）、比特率、PSNR、BD 测度［E79，E80，E94］和计算时间作为指标来比较 HEVC（h265. net）和 H. 264/AVC（用 JM 软件）的性能。请使用 WD 8. 0［E58］。

P. 5. 8 在文档 JCTVC-G399 r2 中，Li 比较了 HEVC WD4 和 H. 264/AVC 高档次的压缩性能。请基于几个测试序列在不同比特率下用 HEVC WD7 和 H. 264/AVC 最新的 JM 软件实现这种比较。和以前一样，SSIM（附录 C）、PSNR、比特率、BD 测度［E79，E80，E94］和实现复杂度作为指标。2011 年 11 月 21-30 日，瑞士日内瓦，JCT-VC 第 7 届会议（比较 HEVC 工作草案 4 和 H. 264/AVC 高档次的压缩性能）。

P. 5. 9 请访问 J. S. Park and T. Ogunfunmi，"A new approach for image quality assessment"，ICIEA 2012，Singapore，18-20 July 2012. 他们制订出了一种主观指

标（类似于 SSIM）用于评价基于（8×8）2D-DCT 的视频质量。他们认为，与 SSIM（指标 C）相比实现该指标相当容易，而在性能上它接近于 SSIM。请基于各种已有的工具评价这一指标。还要考虑（8×8）之外的（4×4）和（16×16）的 2D-DCT；这一概念可被推广到整数 DCT 吗？DCT 可被 DST（离散正弦变换）取代吗？

P. 5. 10　请访问 Dong and Ngan，“Adaptive pre-interpolation filter for high efficiency video coding”，J. VCIR，vol. 22，pp. 697-703，Nov. 2011。Dong 和 Ngan [E14] 设计了一个自适应预内插滤波器（APIF：adaptive pre-interpolation filter）后接一个标准的内插滤波器 [E69]。他们已经将 APIF 集成到 VCEG 的参考软件 KTA 2.6 并且已经与不可分离的自适应内插滤波器（AIF，Adaptive Interpoation Filter）和自适应环路滤波器（ALF，Adaptive Loop Filter）进行了对比。他们使用各种高清序列已经证明，以很低的复杂度 APIF 的性能超越 AIF 或 ALF，而且与 AIF + ALF 可以比拟。请实现 APIF 并证实他们的结论。

P. 5. 11　请访问 Ding 等人”Fast mode dependent directional transform via butterfly-style transform and integer lifting steps”，J. VCIR，vol. 22，pp. 721-726，Nov. 2011 [E14]。他们通过整数提升步骤提出了一种面向快速 MDDT 的新设计。这个方案能够显著降低 MDDT 的复杂度，而编码性能方面的损失可忽略不计。请用整数提升步骤开发（4×4）和（8×8）的 MDDT 并利用视频测试序列与 DCT 和 BSTM（蝶形变换矩阵）进行性能比较。

P. 5. 12　请访问 Li et al.“Compression performance of high efficiency video coding (HEVC) working draft 4”，IEEE ISCAS，pp. 886-889，Seoul，Korea，May 2012 [E20]。他们已经用各种测试序列比较了 HEVC（WD4）和 H.264/AVC（JM 18.0）的性能。他们认为 WD4 对于随机访问应用节约了大约 39% 的比特率（对于相同的 PSNR）、用于低时延场合节约 44%、用于所有的帧内编码节约 25%。请重复这些测试。

P. 5. 13　请访问论文 E. Alshina，A. Alshin and F. C. Fernandez，“Rotational transform for image and video compression”，IEEE ICIP，pp. 3689-3692，2011（见图 5.14）。

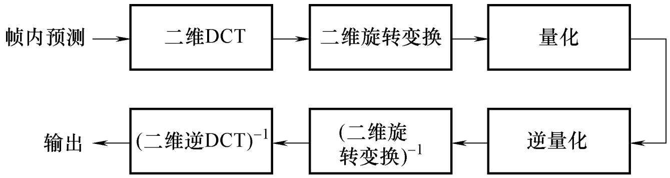


图 5.14　仅用于帧内预测残差的 DCT/ROT 框图

Alshina，Alshin 和 Fernandez 将 ROT4 用于 4×4 的块，将 ROT8 用于所有其他情况下的左上子矩阵（参见论文中的图 2 和图 3），而且表明，

对于所有的测试序列平均起来有 2.5% 的 BD 率增量（参见论文的表4）。请用测试序列实现这个技术并证实以上结果（ROT- rotation transform，旋转变换）。

P. 5. 14　请访问 Saxena 和 Fernandez 提交的文档 JCTVC- C108，Oct. 2010，（题目：Jointly optimal prediction and adaptive primary transform）。他们对于次变换（ROT）打开或关闭的这两种情况下，比较了提出的作为主变换的自适应 DCT/DST 与 16×16，32×32 和 64×64 块的帧内预测中的 DCT。请实现该方案并证实该文档中的表2 和表3 给出的结果。利用 TMuC 0.7。

P. 5. 15　在瑞典斯德哥尔摩的 JCT- VC 会议上，自适应 DCT/DST 已被去掉了。定向 DCT［E110（对于自适应帧内方向预测的残差）也不被考虑。因而方向次变换也不被考虑（见 P. 5. 13）。只有从面向 4×4 大小的亮度帧内预测残差推导出的一个变换，和面向所有其他情况（帧内和帧间）的整数 DCT 被采用了。DDCT 和 ROT（旋转变换）对图像质量贡献及其微小，但代价是在实现复杂度方面需要显著的增加。

参见 Saxena and Fernandez 的论文“On secondary transforms for prediction residuals”，IEEE ICIP 2012，Orlando，FL，2012［E24］。他们将模式相关的 DCT/DST 用于（4×4）大小的块的帧内和帧间预测残差。对于所有其他情况，（即 4×4 之外的帧内和帧间尺寸），他们在主变换 2D- DCT 之后又对左上（低频）系数施加了次变换。与没有进行次变换的情况相比，这种方法对于各种测试序列都导致了 B- D 率的提高（见表 1-3）。请实现这个方案并给出类似表 1-3 的结果。

P. 5. 16　请访问 Zhang and Ma，“Fast intra prediction for high efficiency video coding”，Pacific Rim Conf. on Multimedia，PCM 2012，Singapore，Dec. 2012［E42］.（http://cement. ntu. edu. sg/pcm2012/index. html）

Zhang 和 Ma［E42］提出了一种新的 PU 级的帧内预测方法并实现了 HEVC 编码时间的大幅降低，代价是比特率有可忽略的增加以及可忽略的 PSNR 损失。请实现这个。他们建议，他们的源代码是开源的而且只能用于研究目的（http://vision. poly. edu/ ~zma03/opensrc/sourceHM6. zip）。

P. 5. 17　请参见 P. 5. 16。作者也建议将其他研究者的类似方法（见本文的 5. 2 部分）与他们的工作结合起来进一步降低编码时间。请再参见［E41］和这篇论文后面的参考文献，探索一下。

P. 5. 18　请参见 P. 5. 17。作者 Zhang 和 Ma［E42］也计划探索降低帧间预测模式复杂度的可能性。请研究一下。

P. 5. 19　请参阅 P. 5. 16 到 P. 5. 18。将两种能使 HEVC 编码器实用化的复杂度降减技术（帧内/帧间预测模式）综合起来，并评价压缩性能损失可忽略时 HEVC 编码器复杂度降减的范围。

请注意 P. 5. 17 到 P. 5. 19 都是面向研究的专题，可作为硕士和博士论文的选题。

P. 5. 20　请访问 M. Zhang，C. Zhao and J. Xu，“An adaptive fast intra mode decision in HEVC”，IEEE ICIP 2012，Orlando，FL，Sept. - Oct. 2012［E41］。通过利用来自粗略模式判决的块纹理特征，以及通过残差四叉树分裂过程的进一步简化，他们提出的方法在所有帧内编码高效率测试条件和所有帧内编码低复杂度测试条件下，分别节约 15% 和 20% 的编码时间，同时 BD 率只有微不足道的增加［E79，E80，E94］。请通过实现他们的方法来证实这些测试结果。

P. 5. 21　参见论文 Nightingale et al. “HEVStream；A framework for streaming and evaluation of high efficiency video coding（HEVC）content in loss-prone networks”，IEEE Trans. Consumer Electronics，vol. 59，pp. 404-412，May 2012［E55］。他们设计并实现了一个 HEVC 编码视频流的流媒体和评价框架，并在各种网络条件下测试其性能。报道了使用某些推荐的测试条件（见表 3）时，应用带宽、丢包率和路径时延限制对接收到的视频流质量（PSNR）的影响。实现并证实这些测试。除了 PSNR 之外，使用 SSIM（见附录 C）和 BD（E79，E80，E94）率作为比较目的的基准。

P. 5. 22　参见 P. 5. 21。就未来的工作而言，作者们提出要专注于合适的包/NAL 单元的优先次序方案的发展，该方案用于 HEVC 的选择性丢弃方案。请探索一下，作为能得出进一步结论的研究。

P. 5. 23　参见论文 Marpe et al.，“Improved video compression technology and the emerging high efficiency video coding standard”，IEEE International Conf. on Consumer Electronics，pp. 52-56，Berlin，Germany，Sept. 2011［E56］。代表 Fraunhofer HHI 的作者提出了一种新的视频编码方案，与 H. 264/AVC 高档次相比能节约大约 30% 的比特率，代价是计算复杂度的明显提高。几个对比特率降低有贡献的特征已经被探索过。请实现这个建议并证实带宽的降低。探索一下各种成功用于 H. 264/AVC 编码器（见第 4 章）降低复杂度的技术。这些和其他方法有希望能产生类似于 HEVC 编码器复杂度降低的结果。

P. 5. 24　参见论文 Budagavi and Sze，“Unified forward + inverse transform architecture for HEVC”，IEEE ICIP 2012，Orlando，FL，Sept. - Oct. 2012［E33］。他们利用了 HEVC 的核心变换的几个对称特性并表明，联合实现（嵌入多种块尺寸的变换，在前向和逆向变换之间保持对称，等等）产生比前向和逆向核心变换独立实现低 43%～45% 的面积。他们在图 2 和图 3 中分别给出了统一的前向 + 逆向 4 点和 8 点变换的结构。请开发统一的前向 + 逆向 16 点和 32 点变换的类似结构。注意，这需要导出类似于这篇文章方

程 10-17 所描述的 16 点和 32 点变换的方程。

P. 5. 25 参见 P. 5. 25。作者们声称在前向和逆向变换之间共享的硬件已经能将面积降低超过 40% 。证明这个结论。

P. 5. 26 在转码领域，一些研究者们已经开发、设计、测试和评价过 H. 264/AVC、AVS China、DIRAC、MPEG-2 和 VC-1 之间的转码器。请开发一个在 H. 264/AVC（第 4 章）和 HEVC（主档次）之间的转码系统。使用 HM9。参见［E93］。

P. 5. 27 重复 P. 5. 26，在 MPEG-2 和 HEVC（主档次）之间转码。

P. 5. 28 重复 P. 5. 26，在 DIRAC（第 7 章）和 HEVC（主档次）之间转码。

P. 5. 29 重复 P. 5. 26，在 VC-1（第 8 章）和 HEVC（主档次）之间转码。

P. 5. 30 重复 P. 5. 26，在 AVS China（第 3 章）和 HEVC（主档次）之间转码。

P. 5. 31 如同 H. 264/AVC（第 4 章），HEVC 只包含视频编码。为了用户的可实现性和有用性，需要将音频于 HEVC 的编码视频集成在一起。请在发送端复用编码比特流之后，与音频编码器诸如 AAC、HEAAC 等一起对 HEVC 视频进行编码。保持对口型是接收机的功能，请对两个比特流进行解复用，接着对音频和视频进行解码。对于各种视频空间分辨率和多通道音频实现这些方案。这包括硕士和博士阶段上的几个研究领域。这些集成方案已经面向 H. 264/AVC、Dirac 和 AVS China 与音频编码器完成了实现。

P. 5. 32 类似于 H. 264/AVC 高的视频质量被广播工作室（不是为了传输/分布）所需要，HEVC 帧内编码只能被探索。请在各种比特率下用不同的测试视频比较 HEVC 帧内编码与 H. 264/AVC 帧内编码和 JPEG2000。用 MSE/PSNR/SSIM/BD 率［E79，E80，E94］和实现复杂度作为比较指标。

P. 5. 33 在［E60］，Ohm 等在不同的比特率下，用各种测试视频，以 PSNR 和主观质量作为指标比较了 HEVC 和较早的标准如 H. 262/MPEG-2 视频、H. 263、MPEG-4 视觉（第 2 部分）和 H. 264/AVC 的编码效率。他们也指出，产生所选结果的软件和测试序列可以从 ftp://ftp. hhi. de/ieee-tcsvt/2012/进行访问。

请重复这些测试并验证他们的结果。注意，用于度量主观质量的 DSIS 需要巨大的测试设备、对象（生手和专家）、且可能超出了许多实验室的能提供的条件。

P. 5. 34 用 SSIM（附录 C）和 BD-率［E79，E80，E94］作为性能指标重复 P. 5. 33，并评价这些结果如何与基于 PSNR 的结果进行对比。

P. 5. 35 Horowitz 等［E64］用各种测试序列在不同比特率下比较了 HEVC（HM7. 1）——主档次/低时延配置——和用于低时延的 H. 264/AVC 高档次（JM18. 3）。为了补充这些结果，产品质量级的叫作 x264 的 H. 264/

AVC（第 4 章）编码器与一个产品质量级的来自 eBrisk Video（VideoLAN x264 软件库，http://www.videolan.org/developers/x264.html version 版本 core 122 r2184，2012 年 3 月）的 HEVC 实现软件进行了比较。他们得出结论，对于低时延应用，与 H.264/AVC 相比，HEVC 在平均约为前者 50% 的比特率上一般会产生更好的主观质量。请注意，x264 的配置细节可以从作者那里获得，如果您提出请求的话。有关主观质量/测试的几篇论文在［E44］中被引用。请用 PSNR、BD 率［E79，E80，E94］和 SSIM（附录 C）作为性能指标重复这些测试并评价这些指标如何能与主观质量建立联系。

P.5.36　Bossen 等人［E61］给出了一个详细而全面的有关 HEVC 复杂度（编码器和解码器）的报道并与 H.264/AVC 高档次（第 4 章）进行了对比。他们得出结论，对于类似的视觉质量，HEVC 编码器比 H.264/AVC 编码器复杂好几倍。回报是，HEVC 以 H.264/AVC 所需比特率的一半就能实现与 H.264/AVC 相同的视觉质量。另一方面，HEVC 的解码器复杂度类似于 H.264/AVC 的解码器复杂度。他们声称，手持/移动设备、笔记本电脑、台式机、平板电脑等都能解码并显示编码的视频比特流。这样，实时的 HEVC 解码器就是实用又可行的。他们的优化软件解码器（关于它的优化性没有声明）不依赖多线程且没有用 ARM 和 X64 计算机进行任何的并行化。请在不同的比特率下对于不同的测试序列实现这个软件，并探索进一步优化的其他手段。

P.5.37　列在 FDIS（2013 年 1 月）里的 HEVC 的三个档次之一是只有帧内（图像）编码。请实现这个 HEVC 的编码模式，并以 MSE/PSNR、SSIM（附录 C）和 BD 率［E79，E80，E94］作为指标，与其他图像编码标准诸如 JPEG2000、JPEG-LS、JPEG-XR 和 JPEG（附录 F）进行比较。如以前一样，请在不同的空间分辨率和比特率下用各种测试序列进行这种对比。

P.5.38　除了多视/3D 视频编码之外，可伸缩视频编码（时间、空间和 SNR 质量上）是 HEVC［E59，E68，E73，E74，E117］的扩展/附加内容之一。可伸缩编码（SVC，Scalable Video Coding）目前限于两层（基本层和增强层）。SVC 是 H.264/AVC 的扩展内容之一，而且有关它的专刊已被出版［E67］。SVC 软件在线可得 http://ip.hhi.de/omagecom_GI/savce/downloads/SVC-Reference-software.htm［E68］。请设计、开发并实现 HEVC 中这三种不同的伸缩性。

P.5.39　Sze 和 Budagavi［E65］提出了实现 CABAC（HEVC 的主要挑战）的几项技术。CABAC 能产生更高的信息流量、更高的处理速度和较低的硬件成本，而不会影响高的编码效率。请回顾这些方法的细节并验证这些益处。

P.5.40　［E71］清楚地解释了 HEVC 去块效应滤波器的细节。他们指出，与

H. 264/AVC 相比，这个滤波器除了明显降低视觉瑕疵之外，还具有更低的计算复杂度和多核环境下更好的并行化可能。基于三种配置：（1）全部帧内；（2）随机访问；（3）低时延，他们用测试序列验证了这些结论。请浏览这篇论文和文后引用的相关参考文献，并通过运行仿真来证实这些结果。

P. 5. 41　Lakshman 等［E69］使用最小支撑的最大阶内插来估计运动补偿预测中的分数像素，开发了一个通用的内插框架。与 6 抽头和 12 抽头滤波器［E107］相比，他们的技术显示了改善的性能，特别是对于带有精细的空间细节的序列。然而，这可能会提高复杂度和时延。请开发并行处理技术来降低时延。

P. 5. 42　参见 P. 5. 41。源码、复杂度分析和测试结果 Lakshmanet al.，“CE3：Luma interpolation using MOMS”，JCT- VC D056，Jan. 2011” 可以下载：http://phenix. int- evry. fr/jct/doc_end_user/documents/4_Deagu/wg11/JCTVC- D056-v2. zip. 这是［E69］引用的参考文献 6。请详细地执行这一复杂度分析。

P. 5. 43　Correa 等［E82］研究了 HEVC 编码器的编码效率和计算复杂度。请通过考虑 16 种不同的编码配置集合来实现这一分析。

P. 5. 44　参见 P. 5. 44。证明：低复杂度的编码配置能实现可比拟于高复杂度编码器的编码效率，如草案 8［E58］所描述。

P. 5. 45　参见 P. 5. 43。Correa 等探究的效率和复杂度分析包括的工具（非方形变换、自适应滤波器和 LM 亮度）在后来的 HEVC 草案标准中被去掉了。请通过丢弃这三个工具来进行这个分析。

P. 5. 46　Schierl 等在他们的论文“System layer integration of HEVC”［E81］中建议，HEVC 的运用错误隐藏应该在实现中要仔细考虑，是一个进一步研究的话题。请详细阅读这篇论文并探究 HEVC 的各种抗误码工具。请注意，许多 H. 264/AVC（第 4 章）的抗误码工具诸如 FMO、ASO、冗余分片、数据分裂和 SP/SI 帧由于在现实应用中很少调用而被去除了。

P. 5. 47　请实现由 Zhou 等［E83］提出的 HEVC 主档次无损编码并验证他们的结果。在不同分辨率和比特率下，基于几个测试序列再与当前的无损编码方法诸如 JPEG-2000 等进行比较（参见附录 F）。比较指标是 PSNR/MSE、SSIM（附录 C）、BD 率［E79，E80，E94］等。也要在比较中考虑实现的复杂度。

Cai 等［E112］也针对无损和有损模式对比了 HEVC、H. 264/AVC、JPEG2000 和 JPEG- LS 的性能。对于有损模式，他们的比较只是基于 $PSNR_{avg} = (69PSNR_y + PSNR_u + PSNR_v)/8$。这是针对 4∶2∶0 格式的。基于 SSIM（附录 C）、BD 率［E79，E80，E94］和实现复杂度，扩展这个对比。也要在此比较中包括 JPEG- R，它是基于微软的 HD- Photo 的（见附

录 F)。他们他们提供了一个有关几个标准帧内编码性能比较的范围很广的参考文献列表。另见 P. 5. 37。

P. 5. 48　参见［E93］。通过利用一个对修改的运动矢量重新使用，提出了一个从 H. 264/AVC 到 HEVC 有效的转码器。其中也包含了复杂度可扩展性以平衡为降低复杂度而损失的率失真性能。请实现这个转码器并访问［E93］后面引用的有关转码综述的参考文献（4-7）。

P. 5. 49　参见 P. 5. 48。［E93］的作者建议，H. 264/AVC 的更多信息可被重新用于转码器中来进一步降低转码器的复杂度，这可作为未来的工作。请详细探索这一建议并看看转码器复杂度如何被进一步降低。提出的技术必须基于对比指标进行证明（见 P. 5. 47）。

P. 5. 50　参见 P. 5. 48 和 P. 5. 49。可以开发其他几个转码器，也就是说：

(a) MPEG-2 和 HEVC 之间的转码器（仍然有许多基于 MPEG-2 的解码器）请访问［E95］Shanableh et al. "MPEG-2 to HEVC video transcoding with content-based modeling", IEEE Trans. CSVT, vol. 23, 2013（即将出版）。作者们也开发了一个基于内容基机器学习的有效的转码器。在结论部分，他们提出了未来的工作，请探索一下。在摘要中他们陈述"因为这是首次报道 MPEG-2 到 HEVC 视频转码的工作，所以该报道结果可作为未来转码研究的一个参考基准"这是在转码领域中一个挑战性的研究。

(b) AVS China（第 3 章）和 HEVC 之间转码器。

(c) VC-1（第 8 章）和 HEVC 之间的转码器。

请实现这些转码器。注意，这些研究专题具有硕士论文水平。

您可以访问从网址 http://www-ee.uta.edu/dip 访问已被实现为硕士论文的这些相关转码器，点击课程然后点击 EE5359。或者直接访问 http://www-ee.uta.edu/Dip/Courses/EE5359/index.html.

P. 5. 51　请访问［E70］。这篇论文描述了在标准制订的早期，提交给 HEVC 标准化工作的低复杂度高性能的视频编码。这个提案的一部分被 TMuC 所采纳。这个提案叫作 Tandberg、Ericssson 和 Nokia 测试模型（TENTM）。请实现这个提案并验证结果。TENTM 提案可以从这篇论文最后引用的参考文献 5 中查到。

P. 5. 52　［E70］引用的参考文献 3（也是网址）是指由三星和英国广播公司（在线）提出视频编码技术。请实现这个提案。

P. 5. 53　［E70］引用的参考文献 4 是指由 Fraunhoff HHI（在线）提出的视频编码技术。请实现这个提案。

P. 5. 54　请从 UTA/DIP 网址课程 EE5359 访问 Gangavathi 的硕士论文，题名为 Complexity reduction of H. 264 using parallel programming。通过使用 CUDA，他在基本档次中降低了 H. 264 编码器复杂度的 50%。请把这种方法扩展

到 H. 264（第 4 章）的主档次和高档次。

P. 5. 55 请参见 P. 5. 54。用若干测试序列在不同的比特率下讲 Gangavathi 的方法扩展到 HEVC。用编码器复杂度的降减程度给出性能结果，并以 SSIM（附录 C）、BD-PSNR、BD 比特率［E79，E80，E94］和 PSNR 为指标评价这个方法。UTA/EE5359 课程网址：http://wwwee.uta.edu/Dip/Courses/EE5359/index.html.

P. 5. 56 Zhang、Li 和 Li［E106］开发了一种基于梯度的快速判决算法用于 HEVC 的帧内预测。它既包括预测单元（PU）尺寸又包括角度预测模式。他们声称，与 HM 软件相比，在 HE 帧内编码环境中节约 56.7% 的编码时间，在帧内低复杂度环境中高达 70.86［E95］。请实现这个算法并验证他们的结果。

P. 5. 57 请参见 P. 5. 56。在结论部分，作者们关于结合用在 HEVC 中的率失真优化技术如何对复杂纹理图像获得精确的编码单元分裂。请探索一下。

P. 5. 58 Wang 等［E108］给出了有关多符号比特隐藏方案的研究，HEVC 采用了该方案。这项技术利用数据隐藏方法解决了量化变换系数编码的联合设计问题。他们也给出，这种方法对于所有标准的测试图像总能一致地改善率失真性能，使 HEVC 总体上产生了编码增益。就未来的工作而言，他们认为，通过讲数据隐藏技术应用于其他语法元素，还有指望得到额外的增益。请探索一下这个问题。

P. 5. 59 请参见 P. 5. 58。作者们的意见是，联合量化和熵编码的设计问题保持开放。请探索这个问题。

P. 5. 60 Lv 等［E114］利用在 ARM 处理器上的 SIMD 指令开发了一种快速有效的方法用于运动估计/运动补偿时加速四分之一像素的内插。他们声称，这个方法比基于 HEVC 参考软件 HM 5.2 速度快 5 倍。请参见 5.5 部分的详细加速结果，利用 NEON 技术验证他们的结果。

P. 5. 61 Shi 等［E74］为 HEVC 提出了一个有效的空间可伸缩的视频编码（SSVC，Spatially Scalable Video Coding）方案。他们使用两层帧内预测方案，通过一些测试序列与其他 SSVC 方案相比，他们证明了他们的技术的优越性。在结论部分，他们建议未来的工作要进一步提高他们方法的性能。请详细探索一下这个方案。

P. 5. 62 Zhou 等［E121］通过简单地绕过变换、量化和环路滤波器实现了 HEVC 主档次的无损编码，并且与其他无损编码方法进行了比较诸如，JPEG-2000，ZIP，7-Zip、WinRAR 等。请实现这个方法并与 JPEG、JPEG-XR、PNG 进行对比，考虑将实现复杂度作为另一个指标，在对比中也要卡率 BD 率［E79，E80，E94］和 SSIM（附录 C）。

P. 5. 63 其他文献包括［E124-E127］等提到了将可伸缩视频编码扩展到 HEVC。

请复习这些文献并实现空间/质量（PSNR）/时间的可扩展性。

P. 5. 64　请访问［E64］。在这篇论文中，Horowitz 等说明了使用 HM7. 1 和 JM18. 3 软件，HEVC 以 H. 264/AVC 的一半比特率产生类似的主观质量。用 eBrisk 和 x264 软件也可得出类似结论。请使用最新的 HM 软件，对视频测试序列进行类似的测试并证实这些结果，考虑将实现复杂度作为另一个比较指标。

第 6 章　VP6 视频编码标准

摘要：给出了由 On2 科技公司制订的 VP6 的编码器/解码器细节，强调了 H. 264（第 4 章）和 VP6 之间功能的相似性，指出了二者的差异如黄金帧（golden frames）。也提到了 Google（2010 年 Google 收购了 On2 科技公司）开发的 VP9 相关的开源视频编译码资源。

关键词：VP6　On2 科技　Google　Flash 编译码　黄金帧　无限制运动矢量预测环路滤波　解码　霍夫曼编码器　BoolCoder

6.1　前言

TrueMotion VP6［P3］来自 On2 技术股份有限公司。Adobe 公司批准它作为 Flash 产品套装［P2］，作为 Flash 8 和以后版本的主要编译码的部分。由于它在很高压缩比的情况下给出很好的质量，所以有一些有趣的特征。TrueMotion VP6 是市场上最好的视频编译码系统之一。按照 On2 技术公司，它能提供比视窗媒体视频（Windows Media Video 9）［C16］、Real Video 9［R1］、H. 264［S10］和 QuickTime MPEG-4 更好的图像质量和更快的解码性能。在 On2 技术公司内部进行的测试中，用标准的 MPEG-2 测试视频剪辑，在比较 PSNR 的时候，TrueMotion VP6 能够击败 H. 264 的许多实现方案、Window Media Video 9 和 Real Networks 10。VP6 的视频剪辑更具细节且包含比 Windows Media Video 9 更少的瑕疵，而且比 Real Video 或 H. 264 保持了更多的纹理和细节［P3］。

VP6. 2，TrueMotion VP6 的最新版本，其特征是与以前版本 VP6 相比，性能上有显著的提高［P3］。

6.2　与以前的 Flash 编译码系统 MX 相比

2005 年 Adobe 接受 VP6 并植入 Flash。VP6 能够提供明显好于以前 Flash 编译码系统 MX 的性能，后者使用基于 H. 263 的 Sorenson Spark 编译码系统［S5］。On2 科技公司签发了 VP6 的许可证。［P3］的作者提供了 VP6 编码的 Flash MX 和 Flash 视频的性能对比。这部分描述的结果显示了在多方面都有改进。

这部分用来自［P3］的引用数据和图片描述了来自 On2 科技的作者们的比较研究。图 6. 2 ~ 图 6. 12 所示的图像是时长 12 小时的珊瑚礁探索视频节录。原始的视频由 DV 摄像机拍摄并以照片- jpeg 格式压缩存储［JP6］。因为由 DV 摄像机拍

摄，720×468 的 DV 源需要将某些过扫描部分裁剪掉，而且也要去隔行并改变大小为 320×240［P3］。

用于 VP6 编码的工具是 On2 技术公司的 Flix 专业版。对比提供了 Flix 支持的单通道和双通道编码。这项研究中的样值以相当低的比特率（大约 150kbit/s）进行编码。在图 6.1～图 6.12 中，左边的图像来自 VP6，右边的图像来自 Flash MX 视频。

至于颜色，Flash MX 中的 H.263［S5］实现方案中颜色有过饱和现象。这样导致的视频看起来很“温暖”而且有时候甚至比原始视频刺眼一些。对于 On2，他们的目的很简单：保持原始视频的质量。如果你专门要使视频过饱和从而给它一种锐化的效果，你当然能用编码工具来实现，但这种情况不应该脱离你的控制。饱和的颜色可能会给你一种印象，即某个东西比它实际上更加锐化。如果有怀疑，请放大一幅图像看看实际情况吧（见图 6.1）。图 6.1 并非来自珊瑚礁视频。按照作者，左边的 VP6 图像更加接近与原始图像。

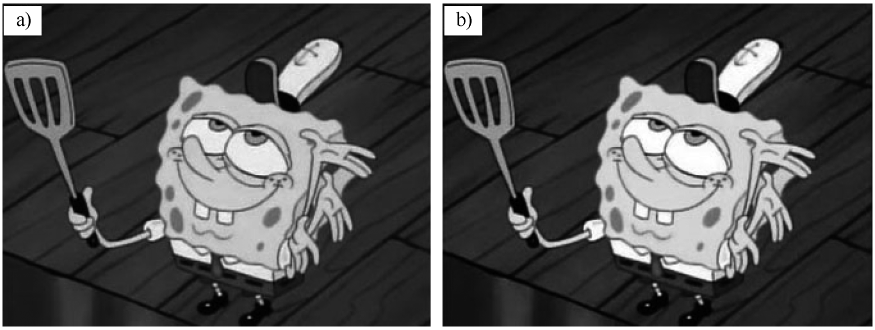


图 6.1 颜色空间比较；忠实于原始图像的 VP6 与过饱和的 MX 版本［P3］

图 6.2 的两幅图像给出 VP6 编码技术与使用 H.263 编码技术的 MX 版本的一种比较。我们可以注意到，VP6 干得相当漂亮，这一点可以从左边的图像看出来；而右边图像的块效应则很明显［P3］。

图 6.3 表明，与一帧内的与 MX 版本相比，VP6 的质量是多么得一致。在 MX 图像中，某些海葵内的尖锐细节得以保持，而更多的微妙细节变得明显斑驳。

图 6.4 的两幅图像再次表明 MX 有时候保持了多么尖锐的细节，对于帧的其余部分造成很大的危害。而左边的 VP6 样例中的鱼和背景在质量上则更加平衡，产生了一幅视觉上更为舒适的图像。

图 6.2～图 6.12 都摘自珊瑚礁探索视频。从图 6.2 和图 6.3 中可以观察到，两种编码的图像都能保持细节；但经过仔细检查就会观察到，MX 图像的许多细节是来自于 8×8 的块效应而不是背景中的细节。从这些图像以及这部分所引用的图像中可以观察到，由于 MX 中编码的块效应，背景区域中的纹理丢失了。图 6.2 和图 6.4中，前景对象的块效应变得显著而清晰。

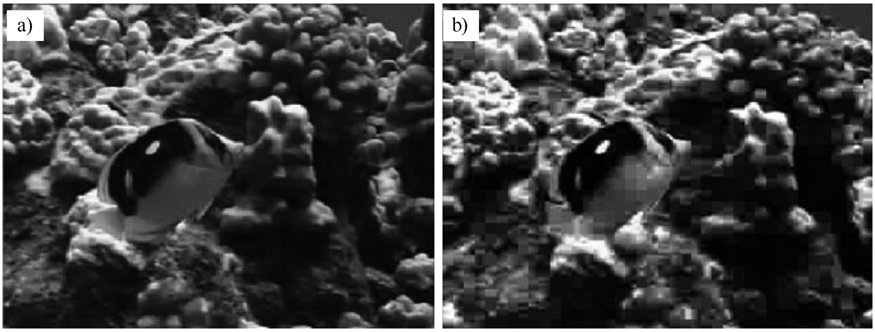


图 6.2　单通道样例：蝴蝶鱼［P3］

a）VP6——较好质量的图像　b）Flash MX——对象和背景的块效应

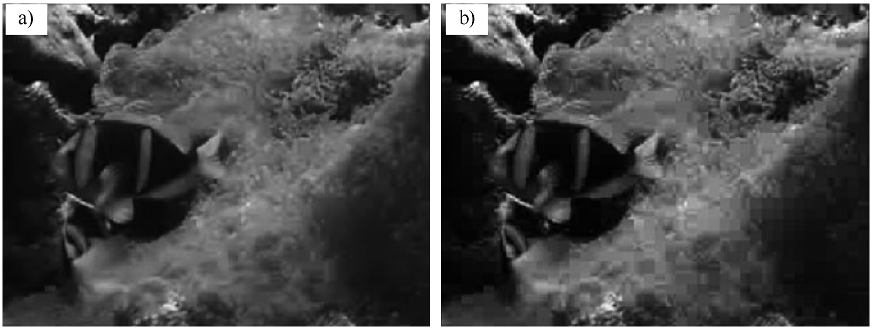


图 6.3　单通道样例：小丑鱼［P3］

a）VP6——较好的质量　b）Flash MX——背景细微细节的丢失

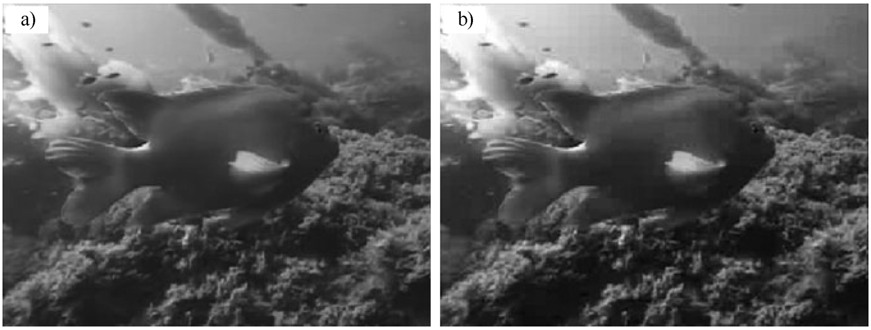


图 6.4　单通道样例：天使鱼，块效应［P3］

a）VP6——较好质量的图片　b）Flash MX——背景和对象中可见的块效应

图 6.5 是关于长须狮子鱼的。在两幅图像中，鱼看起来的美丽程度上有一个明显的区别。从右边的 MX 中观察鱼鳍，可以发现许多因为块效应导致的人工细节。而且，某些信息似乎跳跃到较低的鱼鳍上。颜色明显饱和。除此之外，仔细观察珊瑚背景就会发现，左边的图像比 MX 图像更加真实。

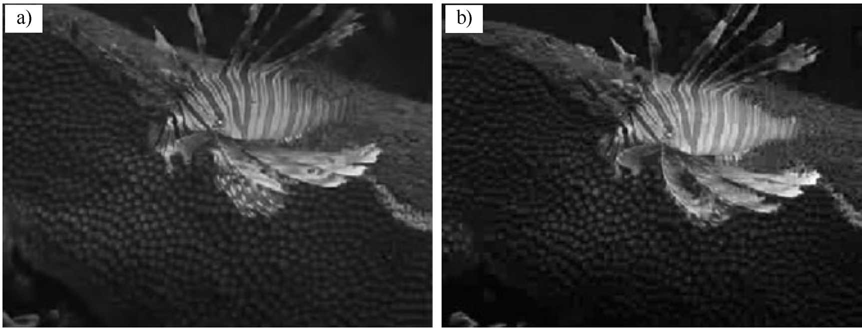


图 6.5　a）VP6　b）Flash MX——可以观察到的人为细节［P3］

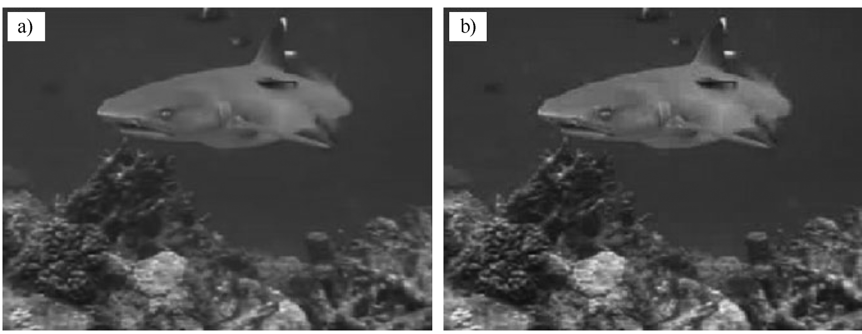


图 6.6　低对比度背景［P3］
a）VP6——清晰而分明的图片　b）Flash MX——质量变坏

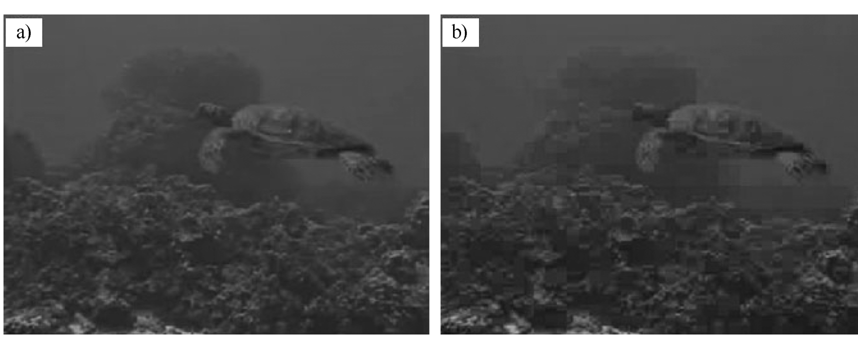


图 6.7　低对比度背景［P3］
a）VP6——图像细节保持了　b）Flash MX——由于块效应背景中的礁石甚至整个丢失了

对 MX 而言，有一个可以观察到的缺点就是不能适应存在低和高对比度的图像。图 6.6 和图 6.7 就是这样的例子；低对比度的海洋背景影响了前景对象（鲨鱼和海龟）。图 6.7 中背景里礁石的情况甚至更差。它几乎隐没到块里面了。

图 6.8 ~ 图 6.12 是双通道编码的例子。双通道编码允许编码器在压缩过程中

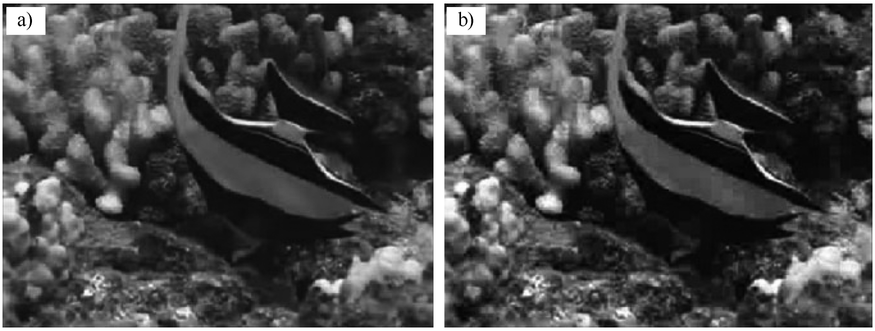


图 6.8 双通道编码［P3］

a）VP6——较好质量的图像 b）Flash MX

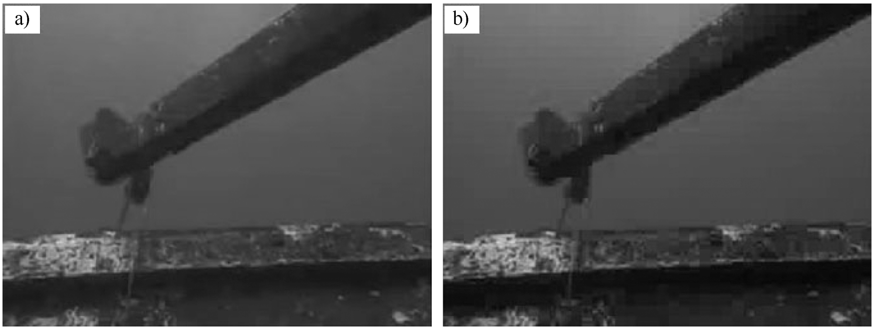


图 6.9 低对比度背景图像（双通道编码）［P1］

a）VP6 b）Flash MX

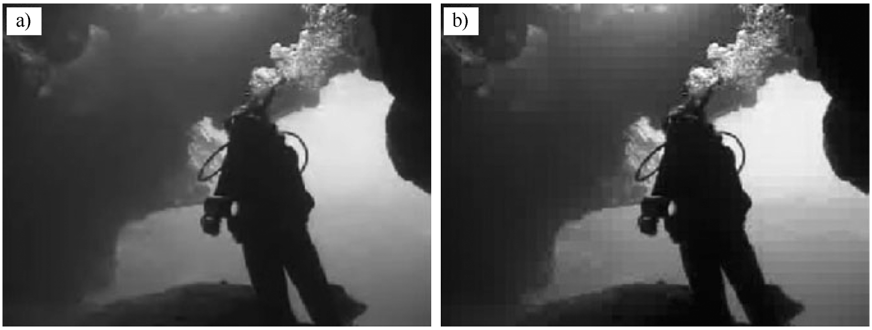


图 6.10 低对比度背景（双通道编码）［P3］

a）VP6——较好的质量 b）Flash MX——有块效应的图像

对于在哪里花费比特做出更好的判决，因而改善了编码的整体质量。

双通道编码的性能比较是明显的。图 6.8 左边的图像呈现地非常好，它甚至在 150kbit/s 上几乎能给人以内容质量很高的印象。

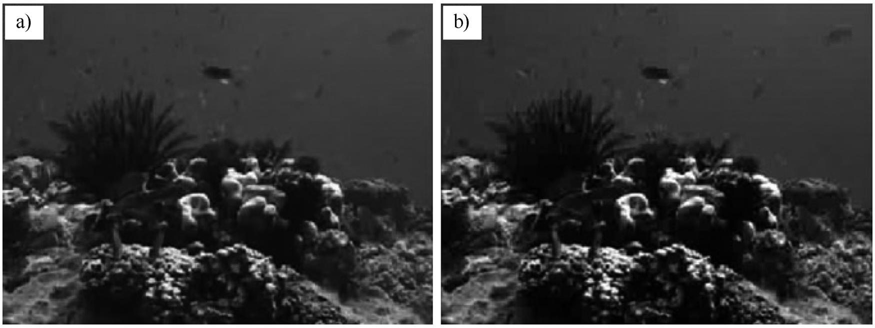


图 6.11　a）VP6　b）Flash MX——背景里的鱼几乎就像低对比度海洋背景中的瑕疵［P3］

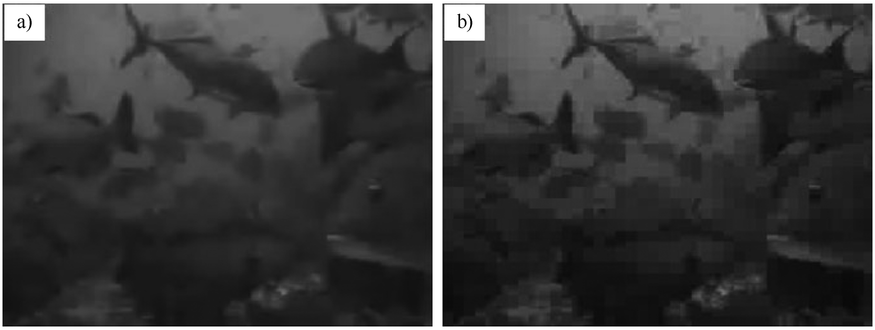


图 6.12　视觉信息的绝对损失（双通道编码）［P3］
a）VP6——保持了质量　b）Flash MX 图像

图 6.9 和图 6.10 是旧 MX 在低对比度背景和高对比度前景时性能差距的例子。与右边背景斑驳的图像相比，两个例子中的 VP6 图像看起来都是很干净的。

图 6.11 是另外一个例子，其中右边 MX 图像的低对比度背景瑕疵和伪明锐的地方几乎损失了小鱼的细节。在图像中他们看起来更像是运动的瑕疵。

最后，图 6.12 总结了研究，给出了 MX 在低对比度场景中与 VP6 性能相比的困难程度。总的来说，可以观察到，与 VP6 相比 MX 性能欠佳，表现为使颜色饱和、不必要地锐化图像且有斑驳的块效应。

6.3　VP6 算法基础

与市场上多数的开源编译码系统一样，VP6 也是一种基于运动补偿和离散余弦变换的编译码系统［P5］。图 6.13 给出了 VP6 编码器的近似框图。图 6.4 ~ 图 6.11部分解释了框图中的各种编码工具和技术。有关编译码系统基础的高级概述如下所示（另见表 6.1）［P5］。

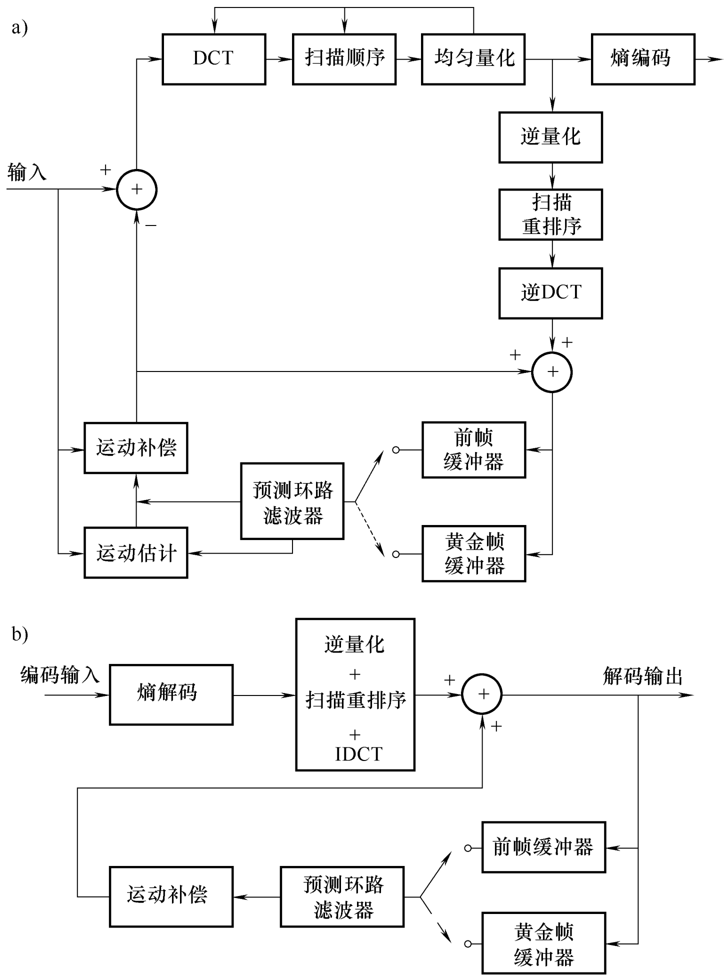


图 6.13　VP6［P8］

a）编码器框图　b）解码器框图

- YUV 4∶2∶0 图像格式；
- 基于宏块的编码（宏块是 16×16 的亮度块加上两个 8×8 的色度块）；
- 1/4 像素精度的运动补偿预测；
- 8×8 的 DCT；
- 64 级均匀量化器；
- 预测环路滤波器；
- 帧可变量化等级；
- 解码后的输出缩放；
- 两种熵编码粗略：霍夫曼和二进制算术（BoolCoder）编码；
- 泛基于内容的熵编码策略。

表 6.1　VP6 和 H.263 编码标准的比较［T6］

算法元素	H.263 基本档次	VP6	H.264 基本档次
图像类型	I，P	I，P	I，P
变换大小	8×8	8×8	4×4
变换	DCT	整数 DTC	整数 DCT
内部预测	无	仅直流模式	是
运动补偿块大小	16×16，8×8	16×16，8×8	16×16，16×8，8×16，8×8，8×4，4×8，4×4
块模式总数	4	10	7 种帧间模式 +（9+4）中帧内模式
运动矢量	1/2 像素	1/4 像素	1/4 像素
去块效应滤波	无	是	是
参考帧	1	最多 2 个	多个

（2008 SPIE）

6.4　VP6 的编码档次

对于很大的图像格式，VP6 编译码系统中所用的某些技术需要相当的计算资源，低端或更高端的处理器可能无法满足。所以为了在编译码系统中分配资源和工具以证实平台的合理性，终端用户需要 VP6 定义的两种不同的档次——VP6 简单档次和高级档次［P5］。

每一帧的首部包含一个标志，VP 档次，它指出用以编码该帧的档次。在两种档次里，BoolCoder 用于在第一数据部分对块和宏块的编码模式以及运动矢量进行编码。

在简单档次编码中 DCT 系数在第二数据分区进行编码，通过设置帧首部的 MultiStream 标志在比特流中加以指示。而且，为了降低计算复杂度，预测环路滤波器和双三次预测滤波器都被禁用。

在使用高级档次时，第二部分是可选项，取决于帧首部中的 MultiStream 标志。如果未被选中，所有的编码数据作为单一部分出现，使用 BoolCoder 进行编码。第二部分可能使用霍夫曼或 BoolCoder 熵编码方案进行编码。此外，预测环路滤波器的使用可以任意激活，取决于帧首部的一个标志，而且预测滤波器类型可以在双线性和双三次变体之间动态切换。

在两个档次的任何一个中，第二部分都是存在的，帧首部中 Use Huffman 标志指明了该数据的编码是用 Huffman 还是 BoolCoder 熵编码方案［P5］。

6.5　帧类型

VP6 只定义了两种帧类型，帧内编码帧和帧间编码帧［P5］。Intra 或 I 帧，像其他任何编译码系统一样不使用参考帧进行重建。由于 I 帧是不需要以前解码帧的

地方，所以它们提供了一种快速随机访问的方法。I 帧没有用到 H. 264 那样的涉及计算复杂度很高的自适应帧内预测［S10］。VP6 所用的帧内编码技术在 6. 6 节进行解释。

帧间预测帧或 P 帧，关于序列中以前编码的参考帧进行差分编码。图 6. 14 描述了以前帧的预测。这个参考帧可能或者是序列中紧邻前帧的重建帧，或者是一幅存储的前帧，叫作金色帧［P5］，如 6. 5. 1 节所描述。

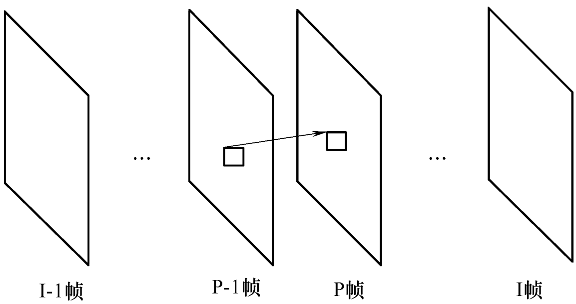


图 6. 14　以前帧的预测

6. 5. 1　黄金帧

备选预测或者叫作黄金帧，是一个帧缓冲器，默认保存最近解码的 I 帧，但它可在任何时间更新。图 6. 15a 和 b 给出黄金帧的两种场景。帧首部的标志为解码器

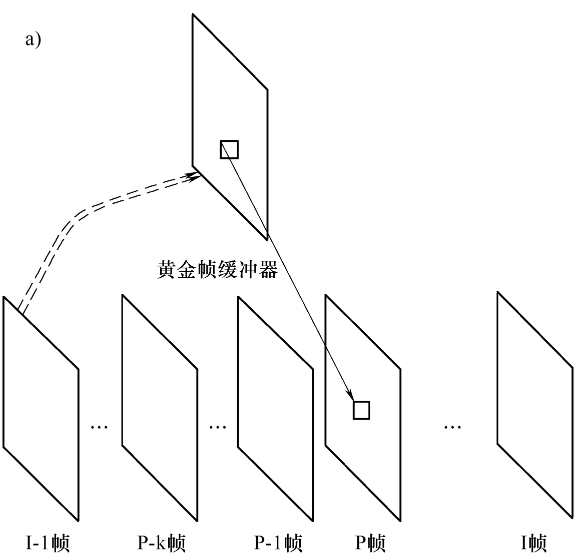


图 6. 15　黄金帧预测　黄金帧缓冲器默认存有最近解码的 I 帧。
黄金帧缓冲器被更新［P5］（2006 On2 科技公司）

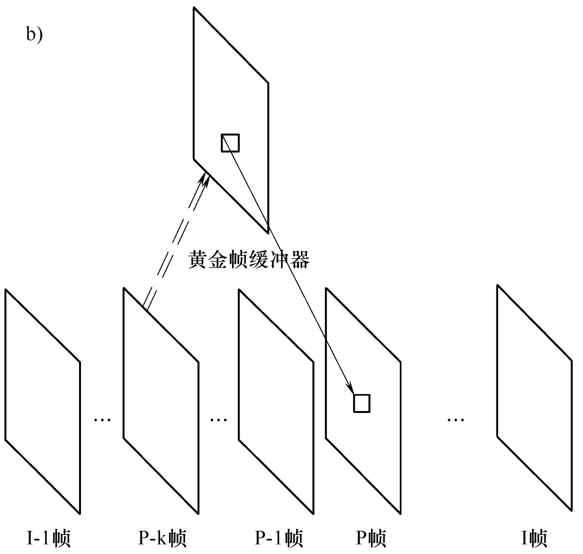


图 6. 15　黄金帧预测　黄金帧缓冲器默认存有最近解码的 I 帧。
黄金帧缓冲器被更新［P5］（2006 On2 科技公司）（续）

指明要不要更新黄金帧缓冲器。为了更新黄金帧，当前帧首先要解码然后整个复制到黄金帧缓冲器中。

VP6 没有使用后向或双向预测。故不存在 MPEG 或 H. 264 那样的 B 帧［P5］。

6. 6　宏块模式

VP6 有十个宏块信令模式——一个帧内模式和九个帧间模式［P5］。

6. 6. 1　I 帧的宏块模式（帧内模式）

当一帧是 I 帧的时候，只使用帧内模式，所以不需要信令模式。与 H. 264 不同，在 CODE_INTRA 中不使用空间预测（VP6 的帧内模式）。6 个块（4 个亮度块和 2 个色度块）的每一个在从每个样值（原始图像——8bit 的 PCM 像素）减去固定值 128 之后进行 DCT 编码。减去 128 有助于改善 DCT 的精确度。

6. 6. 2　P 帧的宏块模式（帧间模式与帧内模式）

P 帧使用运动补偿，故宏块利用预测帧进行预测。被预测帧可能是以前的帧或者黄金帧。运动矢量的单位规定为 1/4 像素（也就是说，对亮度是 1/4 样值精度对色度是 1/8 样值精度）。

用 P 帧的帧内模式实际上像 I 帧一样。

还有其他 9 种定义的帧间模式。这些模式取决于运动矢量是新计算的还是使用来自于相邻宏块其中之一的。用于预测的相邻宏块分类为近邻块和最近邻块［P5］。

6.7 最近邻块和近邻块

在某些环境中，规定一个宏块与最近邻宏块之一具有相同的运动矢量而不是编码一个新的运动矢量，这样做是相当有效的。因为这个原因，VP6 定义了最近邻运动矢量的概念和近邻运动矢量。这些被定义为遇到的前两个非（0，0）运动矢量，——第一个是最近邻的，第二个是近邻的。相邻块及其顺序如图 6.16 所示。对于标记为最近邻或近邻的相邻块，它们应该用与当前块相同的参考帧进行编码。如果不存在这样的块，那么就是没有定义最近邻和近邻运动矢量。

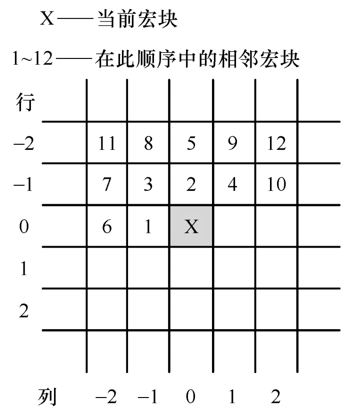


图 6.16 发现近邻和最近邻块的顺序［P5］

不同的编码模式见表 6.2。大多数编码模式是不解自明的。

表 6.2 VP 的宏块编码模式［P5］

编码模式	预测帧	运动矢量
CODE_INTER_NO_MV	以前的重建帧	固定的（0，0）
CODE_INTRA	无	无
CODE_INTER_PLUS_MV	以前的重建帧	新计算的运动矢量
CODE_INTER_NEAREST_MV	以前的重建帧	与最近块相同的运动矢量
CODE_INTER_NEAR_MV	以前的重建帧	与近邻块相同的运动矢量
CODE_USING_GOLDEN	黄金帧	固定：（0，0）
CODE_GOLDEN_MV	黄金帧	新计算的运动矢量
CODE_INTER_FOURMV	以前的重建帧	四个亮度块的每一个有相应的运动矢量
CODE_GOLD_NEAREST_MV	黄金帧	与最近块相同的运动矢量
CODE_GOLD_NEAR_MV	黄金帧	与近邻块相同的运动矢量

对于前帧的重建，对于宏块内的所有块也有可能具有不同的运动矢量。在编码模式 CODE_INTER_FOURNV 中，四个 Y 块的每一个将被独立编码，每个块具有一个相关的编码模式，该模式来自于一个缩减的集合，其中排除了帧内模式或者任何一种黄金帧模式。两个色度块的运动矢量通过对四个 Y 运动矢量取平均计算得到（四舍五入）[P5]。

6.8　运动矢量

VP6 具有包括运动预测的 9 种宏块模式。运动矢量或者是明确计算出来的或者是使用一个相邻块的运动矢量。三种模式包括新运动矢量的明确计算［P5］。

- CODE_INTER_PLUS_MV：新运动矢量参考以前的重建帧进行编码；
- CODE_GOLDEN_MV：新运动矢量参考重建的黄金帧进行编码；
- CODE_INTER_FOURMV：对宏块级上可得的块的子集里每个亮度块可能规定一种不同的模式（见表 6.1）。每个以 CODE_INTER_PLUS_MV 模式编码的块会有自己明确编码的运动矢量。

如果最近宏块存在而且或者在当前宏块的左侧（邻居 1）或者在当前宏块的上方（邻居 2）（见图 6.13），那么新运动矢量参考最近宏块的运动矢量进行差分编码。如果不存在这样的块，那么新运动矢量进行绝对编码。

隐含运动矢量可能是没有运动矢量即（0，0），或者来自最近宏块，或者来自近邻宏块这三种情况之一。这些模式如下：

- CODE_INTER_NO_MV：使用应用于以前重建帧的运动矢量（0，0）；
- CODE_INTER_NEAREST_MV：使用应用于以前重建帧，同时来自以前编码的最近宏块的运动矢量；
- CODE_INTER_NEAR_MV：使用应用于以前重建帧，同时来自以前编码的近邻宏块的运动矢量；
- CODE_USING_GOLDEN：使用应用于重建黄金帧的运动矢量（0，0）；
- CODE_GOLD_NEAREST_MV：使用应用于重建的黄金帧，来自以前编码的最近邻宏块的运动矢量；
- CODE_GOLD_NEAR_MV：用应用于重建的黄金帧，来自以前编码的近邻宏块的运动矢量。

最近和近邻宏块在 6.7 部分解释。

6.8.1　编码

运动矢量有一个 x 分量和 y 分量。每个分量可变分类为或者是短矢量或者是长矢量。所以，一个运动矢量可能将 x 分量编码为短矢量，将 y 分量编码为长矢量或者任何这样的组合。长度是单个分量的长度（x 分量或者 y 分量）；它并非整个的

矢量幅值。

- 短矢量定义为一个长度低于 8 个 1/4 像素单位的矢量；
- 长矢量定义为一个长度大于或等于 8 且低于或等于 127 个 1/4 像素单位的矢量。

6.8.2 预测环路滤波

为了创建一个非零运动矢量的预测块，VP6 有一个预测环路滤波器（图 6.13）. 由于它没有传统的环路滤波，这个滤波也帮助降低块效应。预测环路滤波器的双重用法在 6.11 节解释。

6.9 分数像素运动补偿的滤波

VP6 支持使用分数像素运动补偿，对于亮度有最高 1/4 样值的精度，对于色度有 1/8 样值的精度。在非整数的像素位置上使用内插来确定样值。

支持的两类内插滤波为：

- 双线性滤波：用 2 抽头滤波器（见 6.9.1 节）；
- 双三次滤波：用 4 抽头滤波器（见 6.9.2 节）。

在“简单档次”中不允许双三次滤波，所以在所有需要分数像素预测器的情况下使用双线性滤波。

6.9.1 双线性滤波

表 6.3 所描述 1 维滤波器抽头用于亮度分量 1/4 样值精度的双线性滤波。

表 6.3 用于 1/4 样值精度的亮度滤波的双线性（1 维）滤波器抽头

```
BilinearLumaFilters [4] [2] =
{
{128, 0}, //全样值排列的
{96, 32}, //1/4
{64, 64}, //1/2
{32, 96}, //3/4
}
```

表 6.4 所描述的 1 维滤波器抽头用于色度分量 1/8 样值精度的双线性滤波。

在运动矢量具有 x 和 y 方向上的分数分量的情况下，通过在 x（水平）方向上施加滤波来计算一个中间结果。这个中间结果用作另一个通道的输入，该通道在 y（垂直）方向上进行滤波以生成最终的 2 维滤波输出。

表 6.4　用于 1/8 样值精度色度滤波的双线性（1 维）滤波器抽头［P5］

```
BilinearChromaFilters［8］［2］ =
{
{128，0}，// 全样值排列的
{112，16}，// 1/8
{96，32}，// 1/4
{80，48}，// 3/8
{64，64}，// 1/2
{48，80}，// 5/8
{32，96}，// 3/4
{16，112} // 7/8
}
```

6.9.2　双三次滤波

双三次滤波器的抽头需要计算 16 个从 –0.25 到 –1.00 的 α 值。对于每个 α 值，有 8 组系数对应于从 0 到 7/8 的 1/8 像素偏移。这些值仅仅用在 VP6.2 的比特流中。表 6.5 描述了滤波器抽头值。表中第 17 项用于 VP6.1 的比特流［P5］。

表 6.5　用于 1/8 像素内插的双三次（4 抽头）滤波器抽头

```
BicubicFilterSet［17］［8］［4］ =
{{0，128，0，0}，// 全样值排列，A* = –0.25
{ –3，122，9，0}，// 1/8
{ –4，109，24， –1}，// 1/4
{ –5，91，45， –3}，// 3/8
{ –4，68，68， –4}，// 1/2
{ –3，45，91， –5}，// 5/8
{ –1，24，109， –4}，// 3/4
{0，9，122， –3}，// 7/8
}，
{{0，128，0，0}，// A* = –0.30
{ –4，124，9， –1}，
{ –5，110，25， –2}，
{ –6，91，46， –3}，
{ –5，69，69， –5}，
{ –3，46，91， –6}，
{ –2，25，110， –5}，
{ –1，9，124， –4}，
}，
{{0，128，0，0}，//A* = –0.35
```

（续）

```
{ -4, 123, 10, -1},
{ -6, 110, 26, -2},
{ -7, 92, 47, -4},
{ -6, 70, 70, -6},
{ -4, 47, 92, -7},
{ -2, 26, 110, -6},
{ -1, 10, 123, -4},
},
{{0, 128, 0, 0}, // A* = -0.40
{ -5, 124, 10, -1},
{ -7, 110, 27, -2},
{ -7, 91, 48, -4},
{ -6, 70, 70, -6},
{ -4, 48, 92, -8},
{ -2, 27, 110, -7},
{ -1, 10, 124, -5},
},
{{0, 128, 0, 0}, // A* = -0.45
{ -6, 124, 11, -1},
{ -8, 111, 28, -3},
{ -8, 92, 49, -5},
{ -7, 71, 71, -7},
{ -5, 49, 92, -8},
{ -3, 28, 111, -8},
{ -1, 11, 124, -6},
},
{{0, 128, 0, 0}, // A* = -0.50
{ -6, 123, 12, -1},
{ -9, 111, 29, -3},
{ -9, 93, 50, -6},
{ -8, 72, 72, -8},
{ -6, 50, 93, -9},
{ -3, 29, 111, -9},
{ -1, 12, 123, -6},
},
{{0, 128, 0, 0}, // A* = -0.55
{ -7, 124, 12, -1},
{ -10, 111, 30, -3},
{ -10, 93, 51, -6},
{ -9, 73, 73, -9},
```

（续）

```
{-6, 51, 93, -10},
{-3, 30, 111, -10},
{-1, 12, 124, -7},
},
{{0, 128, 0, 0}, // A* = -0.60
{-7, 123, 13, -1},
{-11, 112, 31, -4},
{-11, 94, 52, -7},
{-10, 74, 74, -10},
{-7, 52, 94, -11},
{-4, 31, 112, -11},
{-1, 13, 123, -7},
},
{{0, 128, 0, 0}, // A* = -0.65
{-8, 124, 13, -1},
{-12, 112, 32, -4},
{-12, 94, 53, -7},
{-10, 74, 74, -10},
{-7, 53, 94, -12},
{-4, 32, 112, -12},
{-1, 13, 124, -8},
},
{{0, 128, 0, 0}, // A* = -0.70
{-9, 124, 14, -1},
{-13, 112, 33, -4},
{-13, 95, 54, -8},
{-11, 75, 75, -11},
{-8, 54, 95, -13},
{-4, 33, 112, -13},
{-1, 14, 124, -9},
},
{{0, 128, 0, 0}, // A* = -0.75
{-9, 123, 15, -1},
{-14, 113, 34, -5},
{-14, 95, 55, -8},
{-12, 76, 76, -12},
{-8, 55, 95, -14},
{-5, 34, 112, -13},
{-1, 15, 123, -9},
},
```

（续）

```
{{0, 128, 0, 0}, // A* = -0.80
{-10, 124, 15, -1},
{-14, 113, 34, -5},
{-15, 96, 56, -9},
{-13, 77, 77, -13},
{-9, 56, 96, -15},
{-5, 34, 113, -14},
{-1, 15, 124, -10},
},
{{0, 128, 0, 0}, // A* = -0.85
{-10, 123, 16, -1},
{-15, 113, 35, -5},
{-16, 98, 56, -10},
{-14, 78, 78, -14},
{-10, 56, 98, -16},
{-5, 35, 113, -15},
{-1, 16, 123, -10},
},
{{0, 128, 0, 0}, // A* = -0.90
{-11, 124, 17, -2},
{-16, 113, 36, -5},
{-17, 98, 57, -10},
{-14, 78, 78, -14},
{-10, 57, 98, -17},
{-5, 36, 113, -16},
{-2, 17, 124, -11},
},
{{0, 128, 0, 0}, // A* = -0.95
{-12, 125, 17, -2},
{-17, 114, 37, -6},
{-18, 99, 58, -11},
{-15, 79, 79, -15},
{-11, 58, 99, -18},
{-6, 37, 114, -17},
{-2, 17, 125, -12},
},
{{0, 128, 0, 0}, // A* = -1.00
{-12, 124, 18, -2},
{-18, 114, 38, -6},
```

（续）

```
{ -19, 99, 59,  -11},
{ -16, 80, 80,  -16},
{ -11, 59, 99,  -19},
{ -6, 38, 114,  -18},
{ -2, 18, 124,  -12},
},
{{0, 128, 0, 0}, // 用于 VP6.1 系数比特流
{ -4, 118, 16,  -2},
{ -7, 106, 34,  -5},
{ -8, 90, 53,  -7},
{ -8, 72, 72,  -8},
{ -7, 53, 90,  -8},
{ -5, 34, 106,  -7},
{ -2, 16, 118,  -4}
```

6.10　对于无约束运动矢量的支持

VP6 支持无约束运动矢量（UMV，Unrestricted Motion Vector）的概念。这意味着对于一个运动矢量而言，指向一个超越图像边界的预测块是有效的。为了支持这个特征，也为了重复编译码系统的伸缩特征，重建缓冲器在所有的方向上用 48 个样值点进行扩展，如图 6.17 所示。

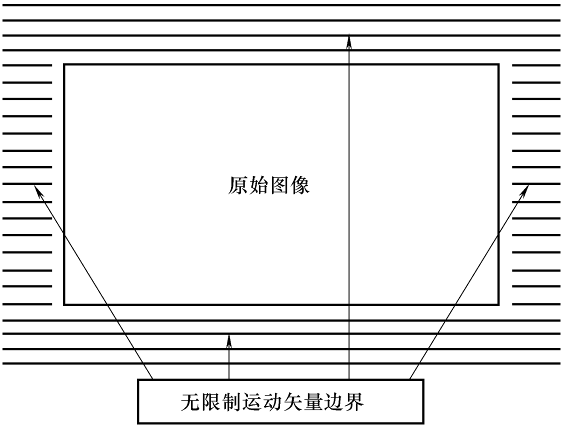


图 6.17　对于超越图像边界的运动矢量的支持

通过复制边缘值 48 次缓冲器得以扩展。首先在 x（水平）方向然后在 y（垂直）方向上进行。

6.11 预测环路滤波

VP6 没有传统上的重建缓冲环路滤波器；但它支持预测帧（以前的重建帧或合适的黄金帧）里 8×8 相邻块边界上的像素滤波，作为对于非零运动矢量创建一个预测块的部分过程。就传统环路滤波器而言，这有助于减少块效应，但滤波在重建缓冲器之内是不能被执行的。更确切地说，输出是复制到一个独立的临时缓冲器中的。对于分数像素运动补偿，需要任何滤波之前都要完成这一步（见 6.7.3 节）[P5]。

在简单档次中，预测环路滤波器是禁用的。在其他档次中，如果帧首部的 Use LoopFilter 标识设置为 1，那么就是启用的。

如果由运动矢量定义的预测块跨越了预测帧内一个 8×8 的块边界，那么就将去块效应和/或去振铃效应滤波器作用于邻近边界的像素上以降低间断性（见图 6.18）。

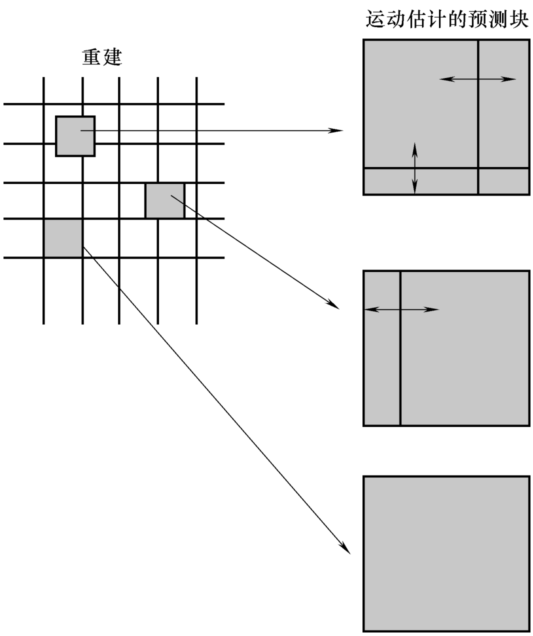


图 6.18 8×8 块边界的预测环路滤波 [P5]

两个滤波器选项如下：

- 去振铃效应滤波器：具有去块效应和去振铃特性；
- 去块效应滤波器：只有去块效应滤波器。去块效应滤波器包括一个 4-抽头滤波器（1，-3，3，-1）以及一个与量化器有关的边界函数，该函数在水平和垂直块边界处发生作用。预测环路滤波器的系数如表 6.6 所描述 [P5]。

表 6.6　预测环路滤波器的限定值［P5］

```
PredictionLoopFilterLimitValues［64］ =
{
30，25，20，20，15，15，14，14，
13，13，12，12，11，11，10，10，
9，9，8，8，7，7，7，7，
6，6，6，6，5，5，5，5，
4，4，4，4，3，3，3，3，
2，2，2，2，2，2，2，2，
2，2，2，2，2，2，2，2，
1，1，1，1，1，1，1，1
}
```

6.12　DCT，扫描顺序和系数牌集合

为了降低解码器的复杂度，VP6 使用了一个略微修改过的非标准定点整数逆离散余弦变换（DCT）；14 比特精度的 IDCT 用于把系数转换回或者差分像素值。该变换基于 Vetterli 和 Ligtenberg 的论文“A Discrete Fourier-Cosine Transform Chip” IEEE Journal on Selected Areas of Communications，vol. 4，pp. 49-61，Jan. 1986［I3］。该论文的作者们提出了一种 DCT 技术，其重点是可变复杂度算法（VCA，Variable Complexity Algorithm），它可以随着所用的目标量化函数而调整前向 DCT 的复杂度。当量化比较粗糙的时候，这一技术能提供较快的性能。生成零或小幅值系数所需的计算可以省略而不会导致严重后果，如果这些系数的位置已知的话。对于逆 DCT 的情况，这使得基于零系数位置的变换及量化数据块的直接分类成为可能。零系数的分组也使得我们具有一种复杂度降低的 IDCT 算法。

这里，前向 DCT 需要能够准确预测量化 DCT 输出的稀疏性，同时具有最小的复杂度开销。这必须在变换和量化实施之前完成。因此，该算法考虑了量化级别和输入块特性［I3］。

能把非零系数分组并放在组的前端，在这种尝试中，自定义 DCT 系数的扫描顺序是有可能的（6.12.1.2）。

6.12.1　扫描顺序

扫描排序是提供自定义扫描顺序的过程。如果我们对 8×8 变换块的 64 个系数以光栅顺序进行编号，那么 0 号和 63 号系数分别就是直流和按频率增加顺序的交流系数，扫描重排序由给出新排序的 64 个元素阵列规定。系数以修改的顺序出现在比特流中。解码器在逆量化和 IDCT 之前再将他们安排回到光栅顺序［P5］。

1. 默认扫描顺序

默认扫描顺序是标准的 Z 形顺序，如图 6. 19 所示。

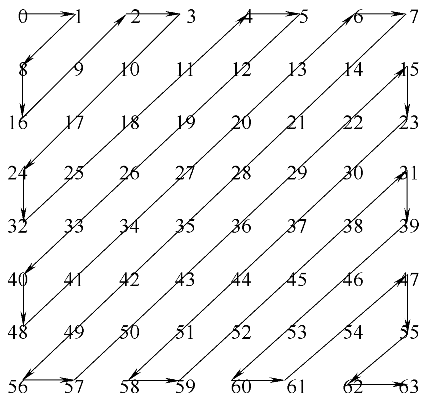


图 6. 19　默认的 Z 形扫描顺序［P5］

2. 自定义扫描顺序

VP6 支持每一帧自定义的扫描顺序。编码器指明使用了自定义的扫描顺序。

对于帧内编码帧扫描顺序首先设置为适当的默认顺序。然后，该默认顺序利用比特流中编码的 Δ 信息进行更新。对于帧间编码帧，Δ 用于前帧所用的自定义顺序中而不是用于默认的扫描顺序之一。

为了规定一个自定义扫描顺序，每个交流系数（在 Z 形顺序中）都指定给一个系数带，如表 6. 7 提到的那样。该表描述了 16 个系数带，调整的扫描顺序中 63 个交流位置分裂到其中。然后，在每个带中的系数基于最初的 Z 形扫描顺序被升序排列。解码器同样保留着表 6. 7，用于对标准 Z 形顺序中每个交流系数的系数带更新信息进行解码［P5］。

表 6. 7　系数带更新概率［P5］

```
CoeffBandUpdateFlagProbs［64］ =
{
NA, 132, 132, 159, 153, 151, 161, 170,
164, 162, 136, 110, 103, 114, 129, 118,
124, 125, 132, 136, 114, 110, 142, 135,
134, 123, 143, 126, 153, 183, 166, 161,
171, 180, 179, 164, 203, 218, 225, 217,
215, 206, 203, 217, 229, 241, 248, 243,
253, 255, 253, 255, 255, 255, 255, 255,
255, 255, 255, 255, 255, 255, 255, 255
}
```

6.12.2　DCT 编码与系数牌集合

DCT 编码有三个级别——直流系数的预测编码，交流系数的编码，以及直流和交流系数的零游程编码［P5］。

1. 直流预测

在解码器，一个块的直流系数通过将预测值和一个预测误差加起来进行重建。预测误差被编码到比特流中然后解码。预测值从当前帧已经解码的相邻块的直流系数值中计算出来。

对于一个特定的块，至多有两个特定的直接相邻块的直流值贡献给预测。如图 6.20 所示，涉及的两个块是当前块紧邻的左侧和上方的块［P5］。

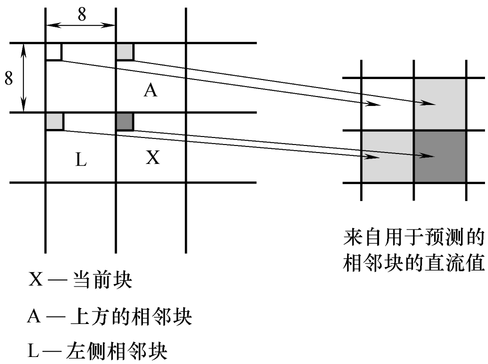


图 6.20　用相邻块的直流预测

相邻块的直流值只对一个特定块的直流值预测有贡献，如果下列所有的条件都满足的话：

- 相邻块存在；在帧的坐边沿没有块的左邻块且在帧的上边沿没有块的上邻块；
- 相邻块从与被预测块（重建的上一帧或黄金帧）相同的参考帧中预测得到；
- 帧间编码块只能由相邻的帧间编码块预测，且帧内编码块只能由相邻的帧内编码块预测［P5］。

表 6.8 描述了预测值是如何在多个相邻块存在，还是一个相邻块存在，还是没有相邻块存在时计算出来的。

表 6.8　基于相邻块的直流预测［P5］

左（L）侧有	上（A）方有	预　测　器
否	否	具有相同预测帧的上一个解码块的直流值
否	是	A
是	否	L
是	是	（L + A + sign（L + A））/2

2. 系数牌集合

表 6.9 描述的下列 12 个牌的集合用来表示量化的 DCT 系数。

表 6.9 DCT 标记集合与额外比特［P5］

序号	系数牌	最小值	最大值	额外比特（包括符号）	额外比特的算术编码
0	ZERO_TOKEN	0	0	*	
1	ONE_TOKEN	1	1	1	B（128）
2	TWO_TOKEN	2	2	1	B（128）
3	THREE_TOKEN	3	3	1	B（128）
4	FOUR_TOKEN	4	4	1	B（128）
5	DCT_VAL_CATEGORY1	5	6	2	B（159），B（128）
6	DCT_VAL_CATEGORY2	7	10	3	B（165），B（145），B（128）
7	DCT_VAL_CATEGORY3	11	18	4	B（173），B（148），B（140），B（128）
8	DCT_VAL_CATEGORY4	19	34	5	B（176），B（155），B（140），B（135），B（128）
9	DCT_VAL_CATEGORY5	35	66	6	B（180），B（157），B（141），B（134），B（130），B（128）
10	DCT_VAL_CATEGORY6	67	2114	12	B（254），B（254），B（243），B（230），B（196），B（157），B（153），B（140），B（133），B（129），B（128）
11	DCT_EOB_TOKEN	N/A	N/A	* *	

表 6.9 中的最小值表示可用那个标志编码的最小数值。额外比特反映了那个标志值的范围。幅值的最高有效位首先发送而最后的额外比特总是符号位。在算术编码中，每个额外比特都以差分概率（原文：differing probabilities——译者注）进行编码。在霍夫曼编码中，这些比特直接进入比特流中。

概率值和上下文信息用于将 DCT 系数编码到这些标志中去。这些概率值存储在表格中，表格由解码器保存并且按帧更新。

在解码器，规定了如图 6.21 所示的直流和交流标志的二进制码树用于解码 DCT 系数牌。码流提供了码树每个结点上取 0 的树枝的概率集。这样，比特流用二进制码树每个结点的零树枝概率来编码，解码器用它来解码 DCT 标志。使用一个解码器能用的算法，同样的概率集也能被转换为一组霍夫曼概率［P5］。

3. 直流解码

对直流而言，解码器保有两个概率集（长度为 11，11 个标志的每一个都有一个值），一个用于 Y 平面一个用于 UV 平面。这些概率集按帧来更新；比特流提供

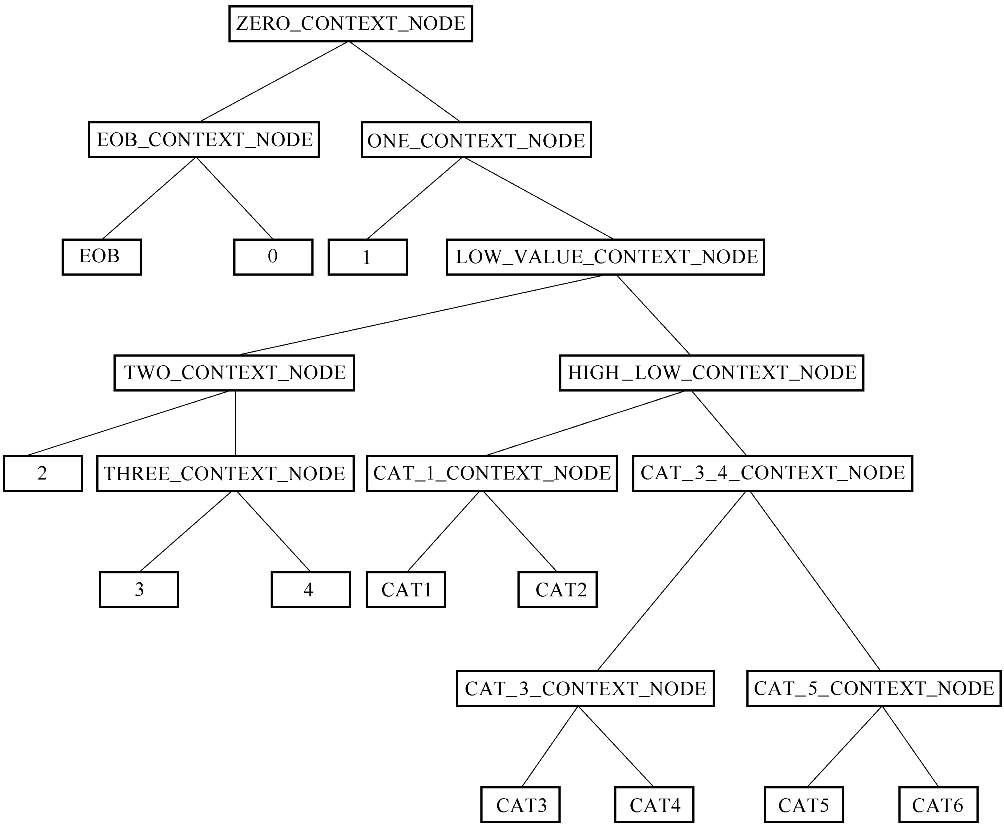


图 6.21 交流和直流上下文的二进制码树

解码器更新这些概率的信息［P5］。

直流系数的算术和霍夫曼解码

算术编码的直流值用更新的概率进行解码。解码直流值要利用上下文信息，这些信息指出当前块的紧邻左侧和紧邻上方的块的直流值是 0 还是非 0（见表 6.10）。

表 6.10 直流结点上下文［P5］

索 引	情 况
0	左侧块的预测直流系数是 0 而且上方块的预测直流系数是 0
1	或者左侧块预测直流系数非零或者上方块预测直流系数非零，但不都是
2	左侧块预测的和上方块预测的直流值

如果已经使用了直流标志的霍夫曼编码，那么解码器就用具备的算法直接从二进制编码码树来生成霍夫曼解码树。然后，这些概率值被用于解码霍夫曼编码的直流值。

4. 交流解码

为了解码交流系数，解码器保持着一个四维的概率集。对于 Y 或 UV 平面，每一个都保持着系数牌的 11 个概率的集合和系数的六个带（原文：band——译者注），如表 6.11 所提到的那样。对于表 6.12 所描述的上下文情况，概率集对于每一个都是不同的。比特流提供解码器更新这些概率的信息［P5］。

表 6.11　交流系数范围［P5］

索　　引	情　　况
0	系数 1
1	系数 2 ~ 4
2	系数 5 ~ 10
3	系数 11 ~ 21
4	系数 22 ~ 36
5	系数 37 ~ 63

表 6.12　前面解码的交流系数上下文［P5］

索　　引	情　　况
0	当前块前面解码的系数（在当前的扫描顺序中）是 0
1	当前块前面解码的系数（在当前的扫描顺序中）是 1
2	当前块前面解码的系数（在当前的扫描顺序中）大于 1

5. 交流系数的算术和霍夫曼解码

在解码器，算术编码的交流系数值用来自比特流中的更新的概率进行解码。所有的四组上下文信息——Y 或 UV 平面，系数所属的系数带，块前面系数是 0，1 还是 >1，以及相应的标志——都需要用来解码交流值。

解码霍夫曼编码的交流系数需要使用 24 个独立的霍夫曼码树，这些码树存放在一个三维数组中。有一个不同的码树，它取决于系数属于哪个平面（Y 还是 UV），块前面的系数是 0，1 还是 >1，以及系数所落入的系数带，见表 6.13［P5］。

表 6.13　霍夫曼编码交流系数范围

索　　引	情　　况
0	系数 1
1	系数 2 ~ 4
2	系数 5 ~ 10
3	系数 11 ~ 63

6. 解码零游程

为了解码零游程解码器必须保有一个二维概率集。零游程概率的第一维由零系数起始的范围来编入索引，见表 6.14。概率表的第二维的索引取决于表 6.15 所描

述的游程值［P5］。

表 6.14　零游程系数范围

索　　引	情　　况
0	系数 1～5
1	系数 6～63

表 6.15　零游程结点索引［P5］

索　　引	游　　程
0	游程概率 >4
1	游程概率 >2
2	游程概率 >1
3	游程概率 >3
4	游程概率 >8
5	游程概率 >6
6	游程概率 >5
7	游程概率 >7
8	比特概率（游程—9）和 1
9	比特概率（(游程—9）>>1）和 1
10	比特概率（(游程—9）>>2）和 1
11	比特概率（(游程—9）>>3）和 1
12	比特概率（(游程—9）>>4）和 1
13	比特概率（(游程—9）>>5）和 1

6.13　量化

视频帧的每一个运动预测的 8×8 块都由编码器进行 DCT 变换到一个 64 个系数的集合。然后，这些 64 个系数通过两个独立的均匀标量量化器进行量化：一个用于直流系数，另一个用于 63 个交流系数。

逆向执行均匀标量量化器需要 64 个系数的每一个乘以整数。直流量化值（被乘数）通过检索直流量化表（见表 6.16）来确定。同样地，交流量化值通过检索交流量化表来确定（表 6.16）［P5］。

表 6.16　直流和交流量化值［P5］

```
DC QuantizationTable［64］ =
{
188，188，188，188，180，172，172，172，
172，172，168，164，164，160，160，160，
160，140，140，140，140，132，132，132，
```

（续）

```
132，128，128，128，108，108，104，104，
100，100，96，96，92，92，76，76，
76，76，72，72，68，64，64，64，
64，64，60，44，44，44，40，40，
36，32，28，20，12，12，8，8
}
AC QuantizationTable [64] =
{
376，368，360，352，344，328，312，296，
280，264，248，232，216，212，208，204，
200，196，192，188，184，180，176，172，
168，160，156，148，144，140，136，132，
128，124，120，116，112，108，104，100，
96，92，88，84，80，76，72，68，
64，60，56，52，48，44，40，36，
32，28，24，20，16，12，8，4
}
```

6.14 熵编码

在 VP6 中，有两种不同的熵编码策略对 DCT 系数标志进行编码——霍夫曼编码器和 BoolCoder [P5]。

霍夫曼编码是一种非常有计算效率的方法，很好地适用于加速优化且具有合理的压缩性能。一般来讲，它在中低级别的处理器上用于很高数据率的情况，因为它比 BoolCoder 能更有效地处理大量的标志。

BoolCoder 是一个简化的二进制算术编码器，允许对标志以分数比特来编码。就压缩效率而言，它比霍夫曼编码器更为有效，但这带来了提高明显的计算复杂度。

霍夫曼编码器和 BoolCoder 两者都使用二进制判决树来表示多比特语法元素。每种情况，码树都被转化为一个树枝判决的序列，且从比特流中读出直到遇到树叶结点。每个树叶结点都有一个相关的语法元素。

两种方案的差别在于树枝判决在树结点的编码方式。霍夫曼编码器用一整个比特来表示一个给定结点上的分支判决。BoolCoder 利用叫作结点概率的概率值。结点概率是给定结点左树枝（零）的概率。所以，BoolCoder 能实现亚比特的判决代价 [P5]。

鉴于霍夫曼编码器完全由二进制判决树确定，BoolCoder 另外需要定义一组结

点概率。结点概率规定为一个数值的数组，这些值按照深度优先遍历树的顺序进行排序。结点概率以线性方式表示为 8 比特：0 代表概率 0，255 代表概率 1。然而，值 0 是明确禁止的。所以，有效范围如下［P5］：

$$1 \leqslant \text{结点概率} \leqslant 255 \tag{6.1}$$

6.14.1　上下文信息的利用

不仅在相邻符号间，而且在各种编码参数之间以及用于编码相邻块的工具之间，都存在着大量的统计相关性。例如，如果最近块以 CODE_INTRA 进行编码，那么就有大约 85% 的可能性当前块也用 CODE_INTRA 进行编码，且只有低至 3% 的可能性当前块用 CODE_INTER_PLUS_MV 进行编码。所以，近邻/最近块的编码模式可被用作上下文信息来对当前块的编码模式进行编码。上下文的这种使用是一种非常有用的方法，它利用了相邻块和像素编码参数之间的相关性，是一种降低比特流中统计信息量的有用途径。通过利用解码器已有的信息，对一组基准概率进行加权以使得他们更好地适应于当前的编码环境。这就实现了更有效的熵编码。

所以从一个所定义的上下文的基准分布得出的条件概率分布就可用于有效的熵编码［P5］。

6.14.2　霍夫曼编码器

为了对霍夫曼编码器编码的语法元素进行解码，霍夫曼解码器遍历了一个特定的二进制码树，每个结点再分枝为左侧子结点或右侧子结点，由比特流中读取的下一个比特（0 表示左，1 表示右）来指明。当遇到树叶结点时遍历结束；每个树叶结点对应一个特定的语法元素。

霍夫曼码树是标准的。该码树用一组树叶结点概率构造。然而，VP6 编码了一组与 BoolCoder 码树编码方式兼容的结点概率，而不是编码树叶结点概率。

所以，解码器需要把比特流中得到的结点概率转化为一组树叶结点概率，以便于它们能够创建霍夫曼码树。后面的解码过程是一个以适当的分枝判决遍历该树的过程。当从树根到树叶结点遍历时，树叶结点概率计算为各个结点概率的乘积，再进行适当的规范化［P5］。

6.14.3　BoolCoder

BoolCoder 基于与二进制算术编码器相同的原理。通过不断地将一个初始的单位区间按照 0 或 1 出现的相对概率比率进行细分从而实现对连续的 0 或 1 进行编码。编码多比特实体可被看作是遍历一颗二进制判决树，树的每个结点都有一个伸出左（或 0）树枝的相关概率。这个概率被称为是结点概率。伸出右（或 1）树枝的概率因而就是 1 减去结点概率［P5］。

6.15 VP6 编码概述

On2 科技公司开发的“真实运动 VP6”（原文：True motion VP6）是市场上具有的最好的编译码系统之一。与 Windows Media 9 video [C16]，Real 9 video [R1]，H.264 [S10] 和 Quick Time MPEG-4 视频编译码系统相比，它提供了更好的图像质量和更快的解码方案。AVS China [A14] 是由中国开发的一款新的经过简化以改善效率的高效视频编码器，它使用了最新的编码工具，用于高清电视内容的编码。AVS 应用领域包括广播电视、高清 DVD 和宽带视频的网络化。因此，把 AVS 编码的信号在互联网上传输变得越来越重要，实现这一目的的可能的方法之一就是开发一个 AVS 到 VP6 的转码器。通过利用现有的参考资料来研究多种转码算法并有效地实现它们从而成功地开发一个计算复杂度降低的转码器，有人已经提交了这样的建议。

VP6 是由 On2 科技公司开发的编码技术。对于现存的 Windows Media Player，Apple QuickTime，和 Real Network Player 在互联网上提供视频业务的播放器，Flash Media 正在作为一种首选的解决方案而出现。在 2005 年，Macromedia 采纳 VP6 作为它的 Flash 播放器中的视频编码算法 [P3]。Flash8 中的 VP6 提供了比现存标准更好的性能，具有更平滑的视频流和低的颜色对比视频图像 [P3]。因此，这就为开发 AVS 到 VP6 转码器创建了一个开端，该转码器把 AVS 编码的信号在互联网上发送。

6.16 VP6 的编码性能

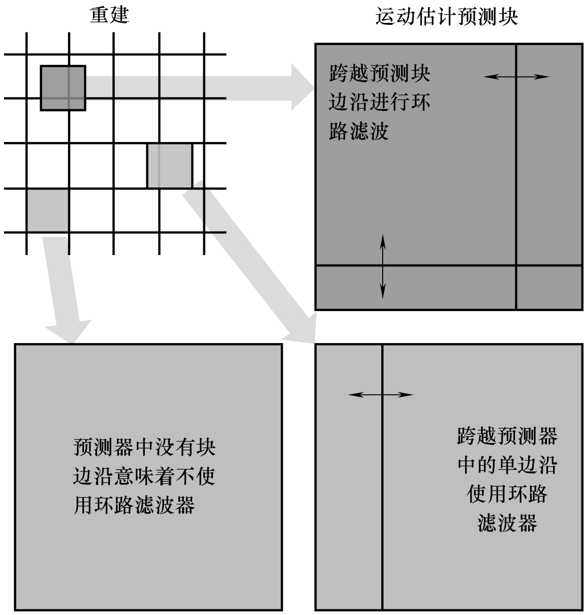
视频压缩器的目的是接收原始的视频并将其压缩到一种更易处理的形式以便于传输或存储。那么，相应的解压器是用于把视频转换回到可以观看的形式。大多数现代编译码器，包括 VP6，都是“有损的”算法，这意味着解码的视频并非精确匹配原始视频。为了达到更高的压缩比，有选择地牺牲了某些信息。编译码器设计者的技术就是在最大化压缩的同时最小化这些损失。

乍一看，VP6 具有与其他主要编译码器有许多共同之处。它用运动补偿来开发时间冗余，用 DCT 来开发空间冗余，用环路滤波器来处理块变换的瑕疵，以及利用熵编码来开发统计相关性。然而，“细节至关重要”（原文：devil is in the details——译者注），所以说，在这一章我们将讨论一些 VP6 的独特之处。

使用基于频率的块变换算法的问题之一是重建视频有时候在块边界上含有视觉上讨厌的不连续性。这些块效应可以通过后处理滤波器进行抑制。然而，这种方法不能解决一个事实，那就是这些瑕疵降低了当前解压帧作为后续帧预测器的价值。

一种备用或补充的方法是在编码器和解码器的重建环路中使用一个滤波器。这种“环路滤波器”平滑了将用于预测后续帧的重建帧缓存器中的块不连续性。大多数情况下这项技术效果不错，但在某些情况下会引起一些问题。首先，对一整幅帧进行环路滤波需要很多 CPU 周期。第二，当在一个图像区域没有明显运动时，在几帧上的重复滤波会导致一些问题诸如抖动。

VP6 对环路滤波采取了不寻常的方法。事实上，有人会说这根本就不说一个环路滤波器而是一个预测滤波器。VP6 不是滤波整个重建帧，而是等待直到跨越块边界的运动矢量被编码为止。只有在这个时候它才复制图像数据的相关块并对通过它的块边沿进行滤波，来创建一个滤波后的预测块（见图 6.22）。



On2's 环路滤波器不仅仅是一种预测滤波器。它通过滤波图像块而不是整个重建图像避免了瑕疵累积，而且只对发生运动的块进行滤波从而节约了CPU周期。

图 6.22 VP6 预测环路滤波器

因为重建缓冲器本身是不会被滤波的，所以没有累积瑕疵诸如抖动这样的危险。另外，因为只有存在明显运动的地方才会使用滤波器，所以这种方法降低了大多数帧的计算复杂度。当 On2（科技公司——译者加）首次在 VP6 中执行这一方法时，它在某些视频剪辑上看到了比传统环路滤波器改善了高达 0.25dB。

6.17 VP6 黄金帧

除了以前的帧外，某些编译码器保持着另外的可被用作预测器的帧。VP6 和

VPx 中的其他编译码器支持一种特殊类型的第二参考帧，我们叫它黄金帧。这个帧可以来自于过去的任意距离（或者至少与前一个黄金帧一样远)，通常以高于平均的质量进行编码。

6.18 背景/前景分割

黄金帧的用途之一是视频中的前景和背景的分割。例如，在多数视频会议应用中背景是静态的。当发言者来回移动时，部分背景临时被遮盖然后又显露出来。通过在黄金帧缓冲器中创建并保持一幅背景的高质量图像，就有可能在它们显露出来时很容易地恢复这些区域。这样就有可能保持背景的质量，即使在前景中有快速运动的时候。而且，节约下来的开销可用于改善整体的编码质量。VP6 编码器也使用黄金帧来改善某些类型的场景质量。例如，在摄像机慢速移动或缩放的时候，由于重复使用环路滤波器或者亚像素运动滤波器会造成细节损失，而周期出现的高质量黄金帧能够通过恢复这些损失的细节提高图像质量。这一高质量帧作为备用的参考缓冲器一直可以使用直到被全部更新。只要运动的速度不是太快，这一帧在更新之后能帮助相当数量的帧稳定图像和改善图像质量。

VP6 编码器通过监视多种因素来确定黄金帧更新的最优频率和质量提升。这些因素包括运动速度、每帧预测下一帧的质量，以及黄金帧被选为编码宏块最佳参考的频度。这个过程的结果对于某些视频剪辑可能是相当引人注目的，如图 6.23 所示。



图 6.23　有（左图）黄金帧与没有（右图）黄金帧的质量改善情况［P7］



图 6.23　有（左图）黄金帧与没有（右图）黄金帧的质量改善情况［P7］（续）

6.19　上下文预测的熵编码

一些其他的高级视频编译码器使用叫作“上下文自适应的二进制算术编码”（CABAC：Context Adaptive Binary Arithmetic Coding）的熵编码技术［H23］。这一技术尽管从压缩的角度来看是相当有效的，但是就 CPU 周期而言却是代价高昂的，因为每当解码一个符号就需要再次计算上下文。VP6 使用一种专有的“上下文预测的二进制算术编码”技术，它依靠在帧级别上复杂的自适应建模。这项技术假定，当考虑当前块特定结果的可能性时，来自空间相关块的信息是相关的。例如，在考虑一个特定 DCT 系数非零的概率时，相邻块中相同系数的信息可能就是很重要的。这里，很重要的一点就是，编码器在帧级别上进行了启发式建模并将相关上下文信息在比特流中传递给解码器。这意味着，在解码器中不必要逐符号（token）地计算上下文。

6.20　比特流分割

为了在建立快速解码器中提供灵活性，VP6 的编码比特流是分段的。所有的预测模式和运动矢量存放在一个数据段中，残差信息存放在另一个数据段。这样，创

建一个预测器帧和解码残差信号的工作就很容易地分开并以最小的开销运行在不同的内核上。通过交替地从一个子比特流（substream）中抽取出模式和运动矢量信息，从其他子比特流抽取宏块的残差信号，VP6 解码器能够一次解码和重建一个宏块。这两种极端之间的任何折中都是可能的，在优化性能和最小化数据与指令缓存丢失的时候提供了最大的灵活性。

6.21 双模式算术与变长编码

除了它专有的“上下文预测的二进制算术编码”算法，VP6 也支持“变长编码（VLC，Variable Length Coding）”。与算术编码器一样，VLC 编码器利用预测的上下文来改善压缩效率。与算术编码方法相比，VLC 方法的效率非常依赖数据率。在很高的数据率上，残差信号中大多数 DCT 系数非零，VLC 编码器与算术编码器之间的区别很小（≤2%）。然而，在低数据率上，算术编码器可能在压缩效率方面给出非常大的改善（>20%）。

因为比特流在一边的预测模式和运动矢量与在另一边的残差信号之间的划分方法，VP6 可以支持 VLC 和算术混合编码。这样，一部分（一般是模式和运动矢量）用算术编码法编码而其他部分用 VLC 方法编码。这使得编码器能以非常有效的方法来平衡解码器复杂度和质量。6.23 部分描述了这种方法如何用在最近发布的 Flash 里的 VP6-S 档次中。

6.22 自适应亚像素运动估计

VP6 的一个很不寻常的特征是，在创建亚像素运动矢量（例如 1/2 和 1/4 运动矢量）的预测块时它使用多个 2 抽头和 4 抽头滤波器的方法。通常，编译码器对所有的块使用单一的滤波器。相反，VP6 支持 16 种不同的 4 抽头滤波器，都具有不同的特征，对 2 抽头双线性滤波器也是如此。编码器可以或者在帧级别上选择一个特定的滤波器，或者按照一个编码器和解码器两者都执行的启发式算法指示该选择应该在 8×8 块级别上实现。这个算法在所选位置检查参考帧的特点并试图为每一块选择最优滤波器，这个块就会既不过分模糊又不过分尖锐。甚至比特流会允许稍微调整滤波器选择算法的参数，所以用户能在编码过程中对于较锐化的视频或者有噪点且斑驳的视频规定一种偏好。提供这一特征是因为认识到对于不同类型压缩瑕疵的态度和接受程度随着不同的人和不同的文化而有相当的不同。

6.23 VP6-E 和 VP6-S 编码器档次

Adobe 公司最近宣称支持 Flash 中的叫作 VP6-S 的新 VP6 档次。这种支持是在

编码端。在解码端，自从 Flash 8 中的 VP6 视频投放以来，VP6-S 和最初档次（VP6-E）已被充分地支持，所以没有后向兼容的问题。两个档次之间的主要差别归结为编码器关于亚像素运动估计、环路滤波器和熵编码所做的判定。如以前提到的那样，VP6 在所有的这些领域都允许相当大的灵活性。

VP6-S 瞄准高清内容，其特征是高的数据率。在这些数据率上，从压缩效率的观点来看，VP6 的“上下文预测的二进制算术编码”编码器和它的“上下文预测的 VLC”之间的差别是不太明显的。然而，高数据率下用在熵解码阶段中的 CPU 周期数大大提高。为了解决这个问题 VP6-S 对残差部分有选择地使用 VLC 方法，如果那个部分的大小超过了一个预定的水平。通过如上所述的 VP6 使用的两个比特流部分，这种折中是有可能的。

此外，VP6-S 限定使用双线性亚像素滤波器，VP6-E 自动为每个宏块选择一个最优的 4 抽头或 2 抽头滤波器。这显著降低了 VP6-S 解码器的复杂度。尽管双线性滤波可能引起某些明锐度和细节的损失，但对于高清视频来说是十分不明显的。质量损失对于较小的图像格式是明显的，使得 VP6-E 称为这种情况下的较好选择。

最后一个重要的区别是 VP6-S 禁用环路滤波器，从而进一步降低了解码器的复杂度。与使用双线性滤波一样，从质量的观点来看此举的不利影响对于高清视频来说是很不明显的。然而，这个区别使得 VP6-S 对于较小的图像格式诸如 QCIF 和 QVGA 很不适应，这时没有环路滤波器会导致视觉质量下降得非常明显。

上面描述的折中方案使得用 VP6-S 档次编码的高清视频有可能在处理能力很低的难以更新的老旧计算机上平滑地回放，而没有太大的质量问题。然而，原始的 VP6-E 档次应该用于较小的图像格式且用在低的数据率下，在此情况下它会传送明显更好的质量。

6.24　设备端口与硬件实现

除了面向基于 PC 的 Windows、Mac 和 Unix 的实现，VP6 已被移植到种类繁多的来自主要公司的设备、芯片组和处理器，如 ARM、德州仪器（OMAP 和 DaVinci）、飞利浦（Philips）、飞思卡尔（Freescale）、漫威（Marvell）、C2、Videantis、索尼（Sony）、雅马哈（Yamaha）和 Archos。而且，On2 发布了一款 VP6 的高度优化的实现硬件［P7］，它在 2008 后半年开始推向市场。这个实现硬件用在移动手持设备的片上系统（SoCs）和其他低功率应用中，也能在移动电话上进行 VP6 视频的高清回放。

谷歌公司在 2010 年 2 月 10 日宣称它已经完成了对 On2 科技公司的收购。继 VP 之后，On2 科技/谷歌已经开发了 VP7—VP10。

VP9/VP10 的相关网址为

http://blog.webmproject.org/2010/06/future-of-vp8-bitstream.html

http://www.webmproject.org/code/repository-layout

http://review.webmproject.org/#change,56

其中有关此时 P9/VP10 最近的东西是 WebM 实验分部，它是开源的且可免费获得。

（承蒙：http://eetimes.com/design/signal-processing-dsp/4017748/The-VP8-video-codec-High-compression-low-complexity）

2008 年秋季宣称的 On2 科技公司第 8 代视频编译码器 On2 VP8，能在比特流中提供压缩性能的显著增益，与它的前一代（VP7）或其他竞争技术如 H.264 相比解压的计算量更低。继承了许多前代版本（VP7 和 VP6）的重大创新，诸如黄金帧、处理器自适应的实时编码和低复杂度环路滤波器，VP8 又增加了 50 多种新技术以很低的复杂度实现低比特率下的杰出性能。

On2 VP8 的设计过程考虑到了实现硬件的广泛性，从 60 MHz 的 ARM9 处理器到今天的高度并行的多核处理器。它在低端机器上实时编码，而且能比其他主要算法花费较少的时间解码。下面这篇文章探索了使得 VP8 工作如此出色的一些创新之处。

J. Bankoski et al.，“Towards a next generation open-source video codec” SPIE/EI，vol. 8666-7，Burlingame，CA，Feb. 2013. 摘要如下：

最近，谷歌公司已经开始开发一种下一代开源视频编译码器——叫作 VP9，作为包含在 WebM 项目（http://www.webmproject.org/）libvpx 库实验分部的一部分。从 2010 年谷歌发布的 VP8 视频编译码器开始作为基准，又加入了许多改善措施和新工具来提高编码效率。这篇文章给出了这个项目当前状态的一个技术总览，以及与其他最新的视频编译码器 H.264/AVC 和 HEVC 的比较。到目前为止，加入的新工具包括：可达 64×64 的更大的预测块尺寸、多种形式的复合帧间预测、帧内预测的更多模式、1/8 像素的运动矢量和 8 抽头可切换亚像素内插滤波器、改进的运动参考帧产生策略和运动矢量编码、改进的熵编码和对不同符号帧级别的熵自适应机制、改进的环路滤波、非对称离散正弦变换和较大的 16×16 和 32×32DCT 的集成、把相似区域分到同组的帧级别分割等等。其他工具和多种比特流特征也在活跃地发挥着作用。VP9 比特流已完成于 2013 年 6 月。结果表明，与主流的最新编译码器相比，VP9 在性能方面相当具有竞争力。

Mukherjee 等用 BD 率作为对比测度［P14］得出结论，VP9 的性能不亚于 HEVC。（29 个 CIF 分辨率和 15 个高清分辨率）测试序列用于执行这一对比。

6.25 总结

本章给出了 On2 科技公司开发的 VP6 编码器细节和 VP9/VP10 的网址。第 7

章将比较 Dirac 视频编译码器与 H. 264/AVC（第 4 章）。

6.26　专题研究

P. 6. 1　Padia［P8］使用运动矢量重用实现了 VP6 到 H. 264（第 4 章）转码器复杂度的下降。请实现它并获得论文中给出的结果。

P. 6. 2　据建议［P8］，通过利用运动矢量细化而不是运动矢量重用，VP6 到 H. 264（第 4 章）转码的视频质量可以进一步改善。请实现之。

P. S. Padia 的毕业论文、PPT 和建议可以从 www-ee. uta. edu/dip 访问。点击课程并点击 EE5359。向下滚动到毕业论文和专题列表。

P. 6. 3　请实现叫作 VP9 的下一代开源视频编译码器（编码器和译码器）并与 VP8、H. 264/AVC（第 4 章）和 HEVC（第 5 章）比较性能。参见［P12］。

P. 6. 4　参见 P. 6. 3。在结论中，Bankoski 等声明，VP9 比特流要在 2013 上半年完成。然后他们期望主办一个有关优化编译码器的速度和质量以及优化特定的应用情况诸如网络上的视频点播和视频会议的 libvpx 库方面的活动。同时，libvpx 库的实验分部（原文 branch——译者）将继续繁荣并在开发新的编码工具方面保持活跃，这些工具将有希望在某一天成为 VP10 的一部分。请探索各种可能的应用中的编译码器优化问题。

P. 6. 5　在论文“Technical overview of VP8, an open source video codec for the web”中，作者 Bankoski，Wilkins 和 Xu（IEEE 关于 ICME 的国际会议，2011 年 7 月）［P13］指出，来自 WebM 项目组的 VP8 参考编码器实现方案仍然没有充分利用论文中所描述的那些 VP8 的全部特征。他们还建议了许多其他技术。因此，在 VP8 编译码器未来的版本中会有很大的创新潜力。请充分探索这一叙述并详细研究，使得 VP8 的压缩性能可被进一步提高。这需要对除 H. 264（第 4 章）之外还要对 VP8 有透彻的理解。

P. 6. 6　P［14］提供了 VP9 技术的简要总览并声称它与 H. 264（第 4 章）和 HEVC（第 5 章）比较具有相当的竞争力。请详细越大这篇论文并证实它的测试结果。VP9 之最新的开源编译码器。请访问 http://www. webmproject. org。

第 7 章　Dirac 视频编译码器与 H. 264/MPEG-4 第 10 部分的性能分析和对比

摘要：由 BBC 开发的 Dirac 视频编译码器不同于其他的标准，因为采用了小波变换而不是整数 DCT。本章列出了小波变换的利弊。除了 Dirac 编码器的细节，还基于 MSE/PSNR、比特率和 SSIM 利用多个测试序列给出了与 H. 264/AVC 的性能比较。

关键词：Dirac　BBC　小波变换　小波滤波器　死区量化　OBMC　DF　压缩比测试　PSNR 测试　SSIM 测试

7.1　前言

视频压缩的目的是尽可能有效地利用有限的存储容量和传输能力，这对于互联网和高清媒体是很重要的。Dirac 是由 BBC 开发的一个开放且免于使用许可税的视频编译码器［D1］［D25］，旨在为从网络视频到高清视频提供高质量的视频压缩，［D20］这样就可以与现存的视频格式诸如 H. 264/AVC［H3，H23，H25，H29］和 SMPTE VC-1 相匹敌。从低分辨率 QCIF（176 × 144 像素）到高清电视（1920 × 1080）以及更高的分辨率，Dirac 可以压缩任何大小的图像，类似于公共视频编译码器诸如 ISO/IEC 运动图像专家组（MPEG）的 MPEG-4 第 2 部分［S9］［H13］和微软的 SMTCVC-1。

Dirac 使用小波变换而不是用在其他编译码器中的离散余弦变换及其整数版本［B12］。Dirac 软件是一款可被自由修改和利用的实现原型。Dirac 的解码器是特别设计得比其他传统解码器快速且更加灵巧。产生的规格实现起来简单又直接，而且进行了优化以提高实时性能［D1］。开源软件诸如 VLC［V11］播放器能解码并显示在 MPEG-2 传输的视频流或 mp4（“. mov”）文件。

除了 C ++ 的 Dirac 参考代码，还有一个叫作 Schrödinger［D20］的高速开源 ANSI C 实现正在积极的开发之中。作为一个 C 库 Schrödinger 是一个 Dirac 视频压缩规格的跨平台实现方案。许多媒体框架诸如 GStreamer［D35］和 ffmpeg［V10］以及诸如 VLC 这样的应用都使用 Schrödinger 来编解码视频。Schrödinger 比 Dirac 参考代码更为优化，而且在大多数编码情况下就编码速度和视觉质量方面的性能更好［D20］。目前 Dirac 实现方案的进展在 diracvideo. org 提供。

Dirac 编译码器的与帧内编码有关的大部分已被批准为 SMPTE 2042（VC-2）中的一个国际标准。Dirac 的这一帧间编码帧版本叫作 DiracPro，［D23］强调质量

和低时延。优化它是为了专业生产、存档应用但不是终端用户分配［D12］。

7.2 Dirac 的体系结构

在 Dirac 编译码器中，图像的运动被跟踪且运动信息被用于进行后面帧的预测。借助运动补偿，对于当前帧和以前帧之间的预测误差施行变换，变换系数被量化和熵编码［D1］。时间和空间冗余分别被运动估计、运动补偿和离散小波变换去除。Dirac 使用了一种叫作算术编码的灵活又有效的熵编码形式，它把比特高效地塞进比特流中［D1］。

7.2.1 Dirac 编码器

视频编码是一个准备视频用于输出的过程，其中的数字视频被编码通过视频编码软件的使用来满足记录和回放所需的合适的格式和规格［D21］。视频流的质量部分地取决于视频编码过程和恰当观看视频所需的带宽。编码视频的时候，对视频和音轨都要进行高度地压缩，以便于它会以这个速度传输。

在 Dirac 编码器［D1］［D21］（见图 7.1）中，整个压缩数据打包在一个简单的字节流中。这种做法具有同步化、允许快速而高效地访问任何一帧——使得编辑简单化。这样的结构使得整个字节流可被打包进许多现存的传输率中。这个特征允许范围很广的编码选项，以及容易地访问所有其他产品制作或广播元数据所需的数据传输系统。

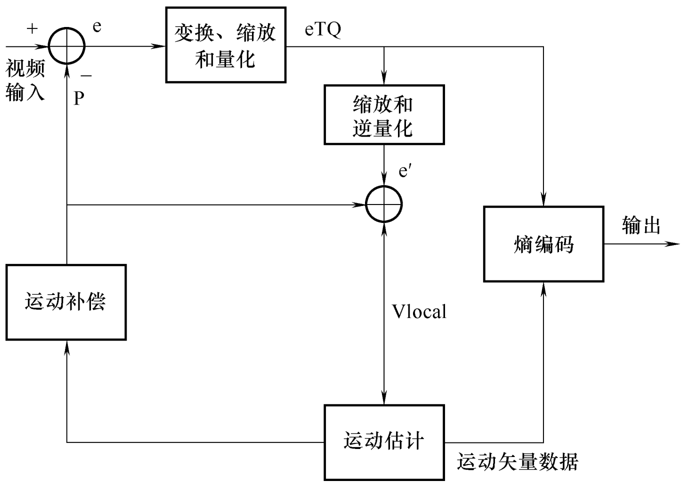


图 7.1 Dirac 编码器的结构［D1］［D25］

在图 7.1 中，每一视频帧输入 Vin 与前一个运动补偿的参考帧 P 进行比较以获得 e，即运动补偿的预测误差（MCPE，Motion Compensated Prediction Error）。eTQ

是经历小波变换、缩放和量化后的 MCPE，而且它辅助熵编码。e′是缩放和逆量化之后的 MCPE。e′与 P（预测帧）相结合可得到 Vlocal，它在运动估计阶段对于生成运动矢量数据是很有用的。P 在每次运动补偿之后进行更新。

7.2.2　Dirac 解码器

Dirac 解码器［D1］［D21］执行编码器的逆向操作。编码数据被熵解码、缩放、逆量化和逆变换来获得重建数据。图 7.2 所示为一幅 Dirac 解码过程的近似框图。Dirac 的解码器实现方案的设计提供快速解码的同时，保持在多种软件平台之间的便携性。

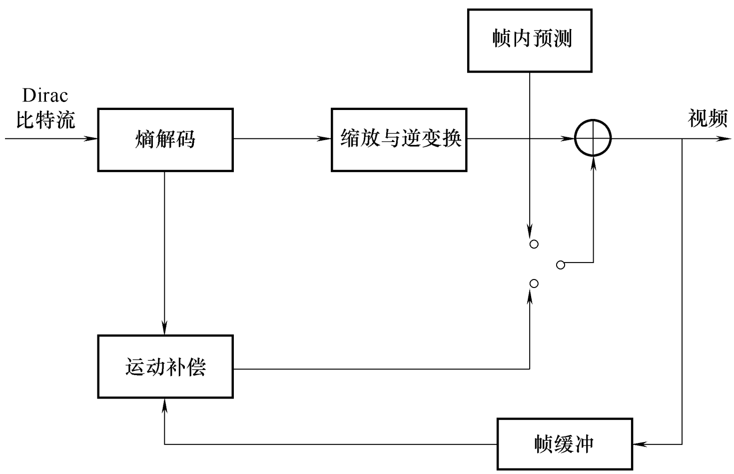


图 7.2　Dirac 解码器结构［D1］［D25］

7.3　Dirac 中的编码和解码阶段

7.3.1　小波变换

2D 离散小波变换（DWT）为 Dirac 提供在一系列分辨率上运行的灵活性。这是因为小波同时工作在整幅图像上，而不是每次集中在小的区域上。在以大致对频率敏感（原文：roughly frequency-sensitive——译者注）的方式解除数据相关方面，Dirac 中的 DWT 发挥着与 MPEG-2 中的 DCT 相同的作用，同时具有保持比基于块的变换更好的细节的优势。合成滤波器能去除由关键采样引入的混叠现象并能完美地重建输入。小波变换通过对信号重复滤波生成低频和高频部分得以实现。对于二维信号，这种滤波在水平和垂直方向都会进行。在每个阶段，水平低频/垂直低频子带会进一步分解，形成了对数频率的子带分解［D20］。

小波变换已被证明，对于静止图像是比块变换更为有效的方法。在 Dirac 小波滤波器中，数据进行三级编码，如图 7. 3 所示。

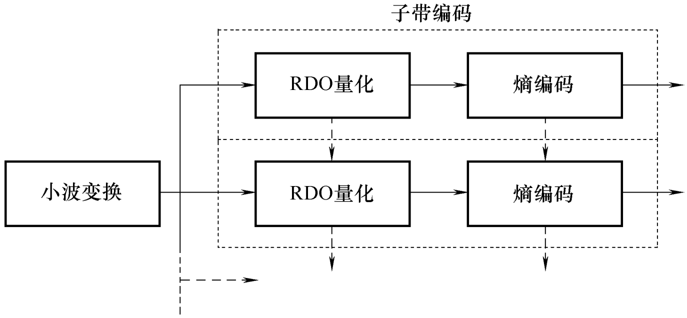


图 7. 3　Dirac 的小波变换框架［D5］（2006 IEEE）

Daubechies 小波滤波器［D18］［D19］用于变换且将数据分解为子带，然后用相应的 RDO（Rate Distortion Optimization，率失真优化）参数进行量化，再进行变长编码。那么，在解码器这三个阶段就要逆过来［D5］。

小波滤波器的选择对于压缩性能是有影响的。为了降低振铃效应和其他影响从而紧致地表示平滑区域，滤波器需要具有紧致的脉冲响应。另外，滤波器对于软件的编码和解码速度也有影响。Dirac 支持数量众多的滤波器来运行在复杂度和性能之间取得平衡。这些在参考软件中是可以配置的［D20］。

Dirac 可用的一种滤波器是 Daubechies（9，7）低通小波滤波器的一种近似，它的提升阶段定义如下：［D20］

$$\begin{aligned} s_n^0 &= x_{2n} \\ d_n^0 &= x_{2n+1} \\ d_n^1 &= d_n^0 - \frac{6497(s_n^0 + s_{n+1}^0)}{4096} \\ s_n^1 &= s_n^0 - \frac{217(d_n^1 + d_{n-1}^1)}{4096} \\ d_n^2 &= d_n^1 + \frac{3616(s_n^1 + s_{n+1}^1)}{4096} \\ s_n^2 &= s_n^1 + \frac{1817(d_n^2 + d_{n-1}^2)}{4096} \end{aligned} \tag{7.1}$$

式中，s 表示和，d 表示差。

式（7. 1）中的数字是 Daubechies 提升系数的整数近似。这使得变换是完全可逆的。这种实现方案忽略了缩放系数，因为这些可以通过适当地对量化噪声加权在量化器选择中加以考虑。这个滤波器的问题是它有四个提升阶段，因此需要软件消耗较长的时间［D20］。

（5，3）Daubechies 高通滤波器则是另一个极端：［D20］

$$d_n^1 = d_n^0 - \frac{(s_n^0 + s_{n+1}^0)}{2} \tag{7.2}$$

$$s_n^1 = s_n^0 + \frac{(d_n^1 + d_{n-1}^1)}{4}$$

DWT 把大多数信息打包在仅仅几个子带（低频）中如图 7.4 所示，这就使压缩成为可能。大多数能量被集中在 LL 子带里面。所有其他的子带可以进行粗糙的量化。



图 7.4　小波变换的几个阶段

这个过程可被重复执行来实现更高级的小波变换。在二维图像的情况下，通常对图像的每个分量在水平和垂直方向上应用小波滤波器以产生四个子带，就是所谓的低-低（LL），低-高（LH），高-低（HL）和高-高（HH）子带。在二维情况下，只有 LL 子带被迭代分解来获得二维谱分解，如图 7.5 所示［D20］。

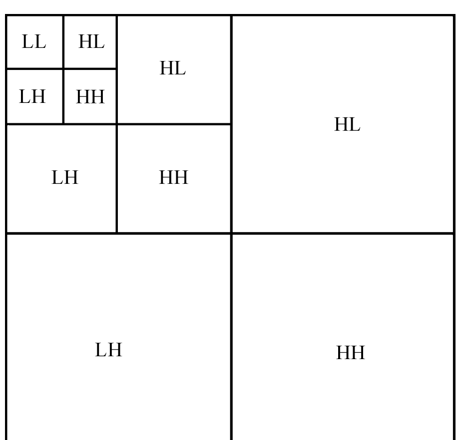


图 7.5　小波变换的频率分解［D5］（2006 IEEE）

7.3.2　缩放和量化

缩放涉及应用小波变换之后接收帧数据并缩放系数来执行量化。量化使用了

一个率失真优化算法来从帧数据中剥离出信息，该算法能产生尽可能少的视觉失真。

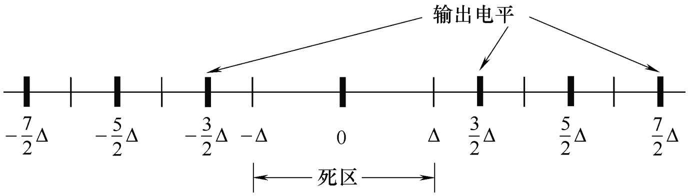


图 7.6　带有质量因数的死区量化器

Dirac 使用不同于均匀量化器的死区量化器技术，第一组量化级步长是其他步长宽度的两倍。该方法简单有效，允许 Dirac 在较小的数值上进行粗糙的量化［D5］。

7.3.3　熵编码

在小波变换之后，实施熵编码来最小化所用的比特数。它包括三个阶段：二进制化、上下文建模和算术编码［D5］，如图 7.7 所示。第一阶段的目的是提供一个易于分析统计分布的比特流，它可用算术编码法来编码。算术编码能自适应于那些统计分布，反映任何局部的统计特征。Dirac 中的上下文建模基于这样的原理：不论一个系数大小，都可以从它的相邻系数或父系数很好地预测出来［D2］。算术编码［DC1］执行无损压缩，既灵活又高效。

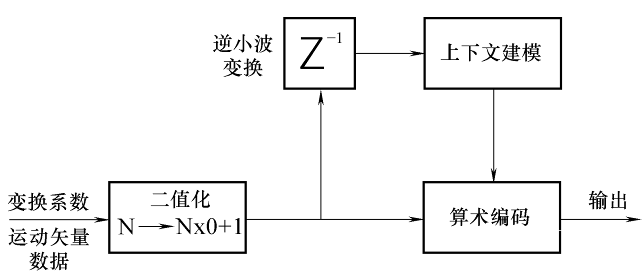


图 7.7　Dirac 的熵编码结构［D5］（2006 IEEE）

小波变换高频子带的非零值常常也在图像的相同部分，就像他们在低频子带那样。Dirac 建立了这些相关性的统计模型，算术编码允许我们利用这些相关性来实现更好的压缩。编码器所估计的运动信息也使用统计建模和算术编码将它压缩到最少的比特数。这些压缩数据被放入比特流中，作为压缩视频的一部分由译码器来使用。

7.3.4　运动估计

运动估计通过查找相邻帧间的相似性来开发视频流中的时间冗余。图 7.8 所示

为用在 Dirac 参考软件中运动估计技术的一个例子。

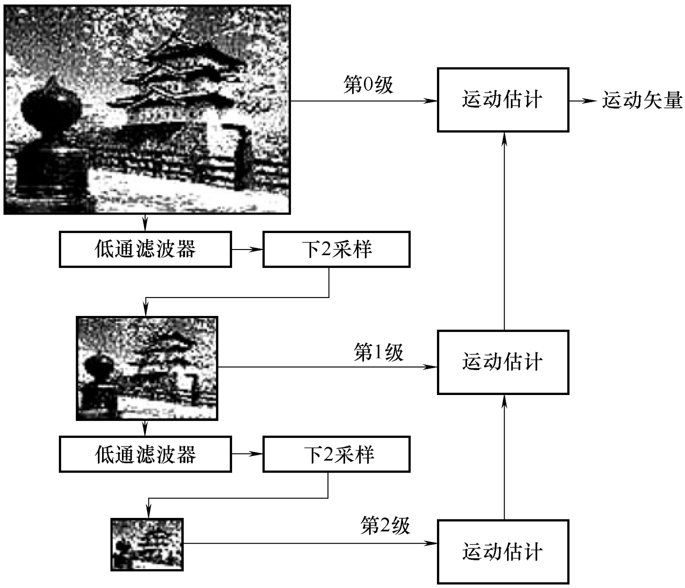


图 7.8　分级运动估计

在第一阶段，通过分级块匹配为每一参考帧和每一块确定像素精度的运动矢量。在第二阶段，通过在紧邻帧中搜索亚像素值对这些像素精度的矢量进行细化。在最后阶段，对每一宏块进行模式判定，确定宏块分裂的级别和每个预测单元的预测模式。这一最后阶段包含进一步的块匹配，因为块运动矢量被用作高级预测单元的候选项 [D3]。

在分级运动估计中，Dirac 首先利用 12 抽头的下转换滤波器 [D8] 转换各种类型的帧间编码帧（既有 P 预测帧又有 B 双向内插帧）的当前帧和参考帧的大小。下转换滤波器是种低通滤波器，允许通过想要的信号并在抽样之前执行抗混叠滤波。任何适当的低通滤波器包括 FIR、IIR 和 CIC 滤波器都可以使用 [D31]。下转换的级数取决于帧格式 [D8]。

Dirac 也定义了三种类型的帧。帧内编码（I）帧不参考序列中的其他帧进行编码。第一级（L1）帧和第二级（L2）帧都是帧间编码帧，也就是说，它们参考以前编码的帧进行编码。

L1 和 L2 帧的区别在于 L1 帧也会用作其他帧的参考，而 L2 帧不会 [D2]。利用标准帧组（GOP，Group of Pictures）结构的帧编码预测结构如图 7.9 所示。Dirac 的每一帧可以从最多两个参考帧中预测。预测模式会随着预测单元而变，且有四种可能性：帧内模式、仅参考 1 模式、仅参考 2 模式、和参考 1 和 2（双向预测）[D3]。

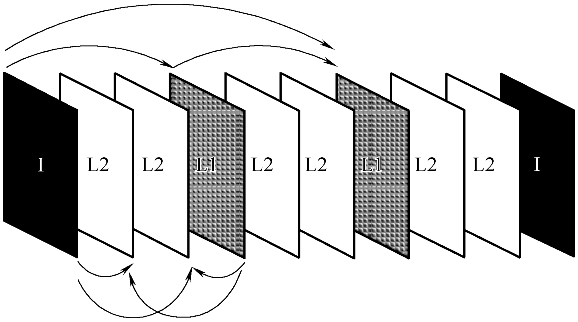


图 7.9　Dirac 的 L1 和 L1 帧［D4］（IEEE 2006）

7.3.5　运动补偿

运动补偿用来预测当前帧。Dirac 利用基于重叠块的运动补偿（OBMC，Overlapped Block-based Motion Compensation）来实现好的压缩性能并避免块边沿效应，用小波编码该效应代价高昂。OBMC 允许相邻块的互动并用基本块来执行，这些基本块被安排进含有一个 4×4 的块阵列的宏块中［D3］。

应该在水平和垂直方向上有一个宏块的精确数目。这是由数据填充来实现的。因为在运动补偿之后进行小波变换，这一过程对于分割性具有自己的要求，故还会需要进一步的填充［D20］。

尽管 Dirac 并非特别设计使其具有可扩展性，但块大小是唯一不可扩展的因素，而且对于较低分辨率的帧可以很容易地选择较小的块。Dirac 的 OBMC 方案基于可分离的线性斜坡掩模。它在预测块上充当一个加权函数。给定第 t 帧的一个像素 $p=p(x, y, t)$，p 可能落在仅仅一个块之内或最多四个块之内如果它位于块角处如图 7.10 所示，其中阴影区域表示重叠区域［D20］。

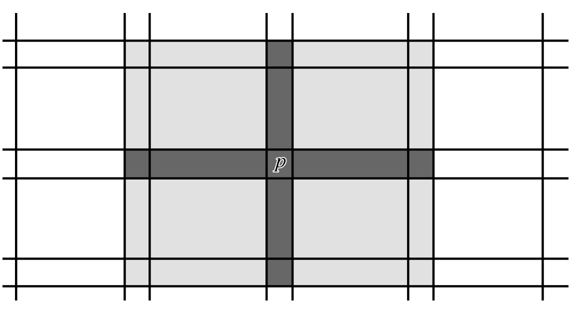


图 7.10　OBMC 的重叠块［D20］

每一宏块可被分成多个预测单元，或者包括 16 个单独的块，或者叫作亚宏块的 4 个中等大小的块的阵列，或者一个宏块大小的单个块（见图 7.11）。

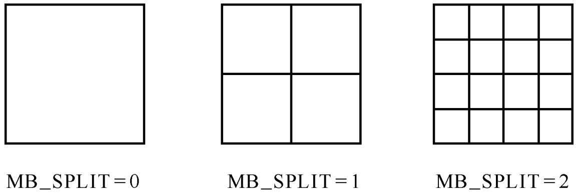


图 7. 11 Dirac 宏块分裂为子块的模式（2006 IEEE）

OBMC 参数可能会随着帧而改变，但也存在基于帧大小的默认值。对于垂直和水平方向（维数对于不同分辨率的色度分量要进行适当的缩放）有 8 个像素重叠的 12×12 的块，存在成流（原文：streaming——译者注）和标清分辨率的默认值。所用的 OBMC 重叠函数是一个上凸余弦函数（原文：raised-cosine function——译者注）的近似［D3］。

Dirac 也提供亚像素运动矢量的运动补偿，因此允许预测率可达 1/8 像素的精度。然而，所用的实际运动矢量精度可能较低，取决于最佳的平衡点，它基本上由所选的比特率来定。也支持一些技术诸如只用运动信息预测一帧，和低比特率下预测一幅几乎等同于前一帧的帧的技术。

7. 3. 6 解码器

解码过程分三个阶段执行，如图 7. 12 所示。在第一阶段，输入的编码比特流由熵解码技术进行解码。接下来，执行缩放和逆量化。在最后的阶段，对数据进行逆小波变换以产生解码的、无压缩的视频输出。在视频质量和运动矢量比特率之间也需要均衡考虑［D5］。

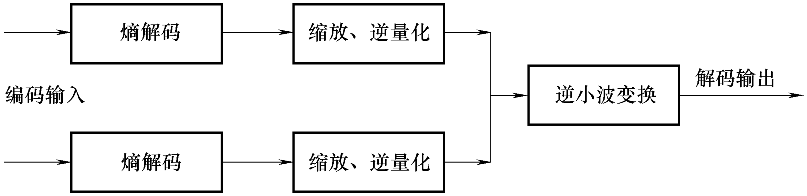


图 7. 12 Dirac 的解码阶段［D15］

7. 4 实现

Dirac 参考软件完全用 C++ 编程语言实现，C++ 允许在所有公共操作系统上进行面向对象的开发。C++ 代码编译后产生公有函数、运动估计、编码和译码的库，这些库具有接口允许从 C 调用。可以用 C 编写应用程序员的接口，从而使得它能保持简单性且能与各种媒体播放器、视频处理工具和流媒体软件集成［D1］。

7.4.1　编码结构概述

Dirac 编译码器具有一个面向对象的编码结构。编码器包括多种实体（原文：objects——译者注）负责图像序列内的特定‘对象’的压缩。换言之，序列、帧和图像分量的压缩都定义成单独的类。

7.4.2　编码的简明性和相对速度

由于 Dirac 参考软件相对简单，人们发现它的编码速度远远快于 H. 264［H3，H23，H25，H29］JM17. 1 参考软件。两种编译码器的解码速度相当。

有相当多的研究论文［D2］［D10］［D28］建议一些技术来优化 Dirac 的熵编码器。按照一篇论文［D28］，通过用精确配置的 M-编码器取代 Dirac 算法中原始的算术编码器，可以实现快得多的视频编译码。这个新算术编码器在高比特率下速度快三倍，甚至性能超越了原始压缩性能。

另一篇论文［D10］对 Dirac 编译码器建议了一种基于质量因数优化的率控制算法。这种方法利用现有的恒定质量控制算法，该算法取决于一个给出固定比特率的叫作质量因数（QF，Quality Factor）的参数。

在 Dirac，总的均衡因数从 QF 得出，意思是质量或量化因数。QF 不是一种质量的直接度量。用固定的 QF 编码会保证只在同源的图像材料具有固定的质量，其中，失真和比特率之间的均衡是固定的［D11］。

图像 λ（原文：lambda——译者注）值用于量化和运动估计的率失真控制：它们最初从图像 QF 中获得，或者设置在命令行上并用于所有的图像，或者通过率失真控制算法来确定。然而，一些因数被用于在运动估计之后修改 λ 值［D11］。

λ 值的初始化如下：［D11］

$$
\begin{aligned}
I_{\text{lambda}} &= \frac{1}{16} \times 10^{(10-\text{QF})/2.5} \\
L1_{\text{lambda}} &= 32 \times I_{\text{lambda}} \\
L2_{\text{lambda}} &= 256 \times I_{\text{lambda}}
\end{aligned} \tag{7.3}
$$

这些 λ 变量用于在 I、L1 和 L2 图像中的量化器选择（图 7. 9）。从这些参数中再得出运动估计的 λ。理想的均衡点会随着不同的序列、视频分辨率、感知权重或块大小而改变［D11］

$$
\begin{aligned}
L1^{\text{me}}_{\text{lambda}} &= 2.0 \times \sqrt{L1_{\text{lambda}}} \\
L2^{\text{me}}_{\text{lambda}} &= L1^{\text{me}}_{\text{lambda}}
\end{aligned} \tag{7.4}
$$

对于 I、L1 和 L2 帧的指导原则如下：［D11］

1）I 帧应该比 L1 帧质量高且 L2 帧应该比 L1 质量高。

2）运动数据和良好的运动显示在较低比特率（低 QF）下比较高比特率（高 QF）下更为显著。

出现第一个原则是因为 I 帧用作 L1 和 L2 帧的参考帧；L1 帧用作 L2 的参考帧。如果质量从 I 上升到 L1 或从 L1 上升到 L2，那么编码器会需要纠正参考帧的量化误差并由运动补偿“向前推送”。这个误差像噪声且编码代价高。还有，当那个帧的质量通过运动补偿改变的时候，参考帧中单个系数的误差可能扩散到多个系数。因此，L1 和 L2 的 λ 相乘。第二条原则的目的是以防止质量发生断崖式下降，因为当 QF 下降时，λ 上升。运动场在低比特率时并非过度平滑。即使质量较低，也没有纠正很差的区域。L2 帧在残差编码时纠正运动估计误差的机会更少［D11］。

人们利用来自前帧编码的比特率导出一个叫作率-质量因数（R-QF，Rate-Quality Factor）的数学模型来生成当前编码帧的最优 QF 以满足目标比特率要求。

在另一个研究项目中广泛讨论了不同的编码器优化方法，例如多线程、单指令多数据流（SIMD，Single Instruction Multiple Data）扩展指令集（SSE，Streaming SIMD Extensions）［D32］和用 Visual Studio 附加程序的英特尔公司的 C/C++ 编译器进行编辑。

7.5　结果

目标测试方法努力确定参考帧和编码比特流之间的误差量［D5］。为了保证测试的精确性，需要保持一个兼容的测试平台。这会需要两个编译码器在同一比特率下进行测试［D5］［D10］。

因为 Dirac 的最新版本包括了一种固定比特率（CBR，Constant Bit Rate）模式，那么 Dirac 和 H.264/MPEG-4 第 10 部分的性能比较通过在不同的比特率下编码若干测试序列来产生。通过利用 H.264 的 CBR 模式，我们可以保证 H.264 正在以与 Dirac 同样的比特率进行编码［D10］。

目标测试分为三个部分，也就是说①压缩，②结构相似度指标（SSIM）和③峰值信噪比 PSNR。测试序列“Miss-America”QCIF（176×144）［V3］，“Stefan”CIF（352×288）［V3］和“Susie”标清（720×480）［V4］用于性能评价。这两种方法在压缩、PSNR 和 SSIM 方面是非常接近且具有可比性的。另外，针对所有的测试视频与 H.264 相比，由 Dirac 实现的编码时间有明显的提高。

7.5.1　压缩比测试

通过得到*.drc 和*.264 文件的大小，再与原始序列文件的大小进行比较，就分别得出了 Dirac 和 H.264 的压缩比。

利用 CBR 模式，有可能对两个编译码器都设置一个“目标比特率”，对 Dirac 而言这比质量也就是说 QF 还重要。这能确保两种编译码器用在相同的工作环境中。在这些测试中 QF 用比特率（kbit/s）指标代替。

图 7. 13 ~ 图 7. 15 分别给出了 Dirac 和 H. 264 在对 QCIF、CIF 和 SDTV 序列进行压缩的性能比较。Ravi ［D14］ 以及 Rav 和 Rao ［D15］ 已经进行了这样一个对比。

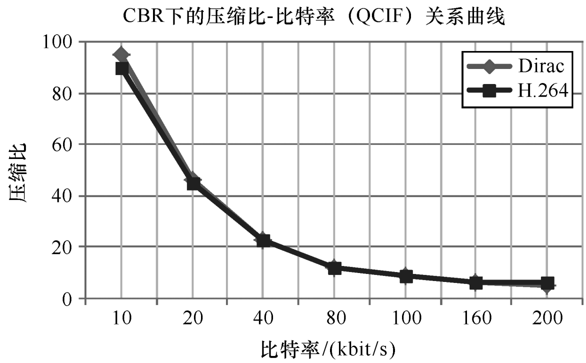


图 7. 13　针对 QCIF 序列 “Miss- America” 的 Dirac 和 H. 264 的压缩比对比 ［D15］

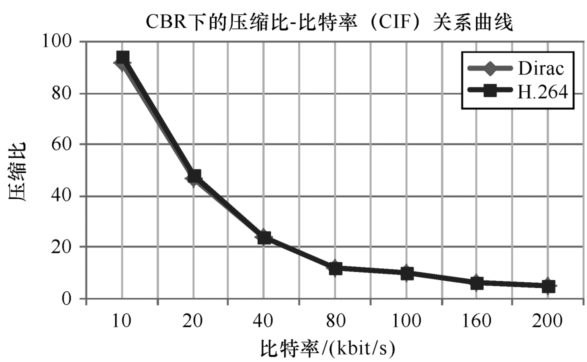


图 7. 14　针对 CIF 序列 “Stefan” 的 Dirac 和 H. 264 的压缩比对比 ［D15］

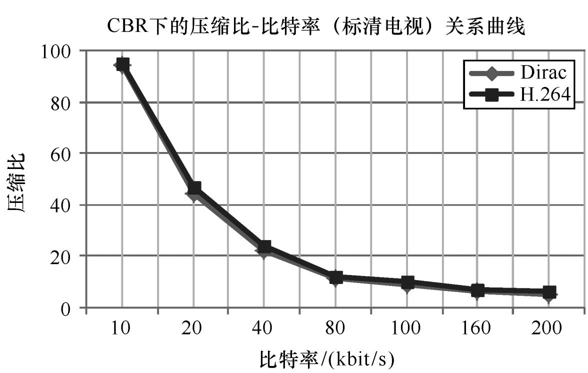


图 7. 15　针对标清电视序列 “Susie” 的 Dirac 和 H. 264 的压缩比对比 ［D15］

针对 QCIF 序列的情况，Dirac 在较低比特率下实现了比 H. 264 略高的压缩比。在较高比特率下，Dirac 和 H. 264 两者都实现了类似的压缩比［D14］。

对于 CIF 和标清视频，H. 264 在较低比特率下给出略好的压缩效果。在较高比特率下，Dirac 和 H. 264 两者都能实现类似的压缩效果。

7. 5. 2 SSIM 测试

结构相似度指标（SSIM）［Q13］借助于比较亮度和对比度已被标准化的像素亮度的局部图案进行工作［Q13］。从根本上说，SSIM 的计算基于亮度相似度、对比度相似度和结构相似度组合成为的一个值。SSIM 的最大可能值是 1，这种情况表明编码的序列就是参考序列的一个精确复制品。SSIM 也是客观评价视频质量的另一种方法［D5］。H. 264 能实现比 Dirac 稍好一点的 SSIM，如图 7. 16 ~ 图 7. 18 所示。

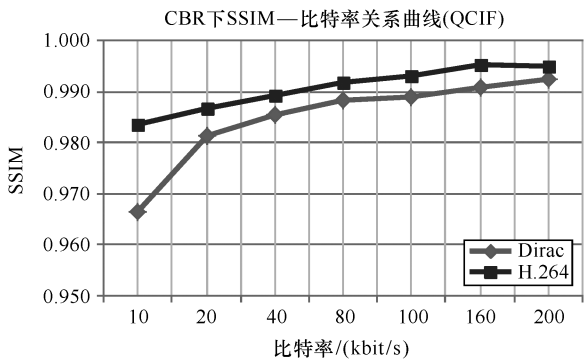


图 7. 16 针对 QCIF 序列“Miss-America”Dirac 和 H. 264 的 SSIM 对比［D15］

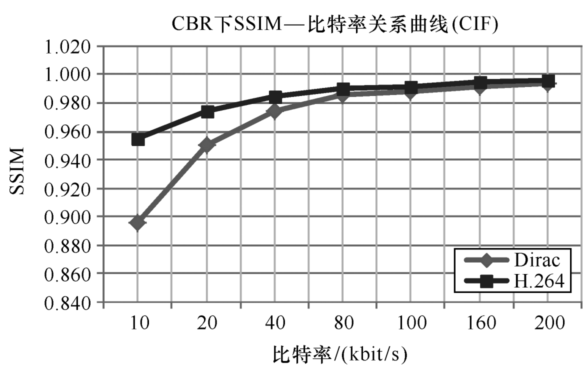


图 7. 17 针对 CIF 序列“Stefan”Dirac 和 H. 264 的 SSIM 对比［D15］

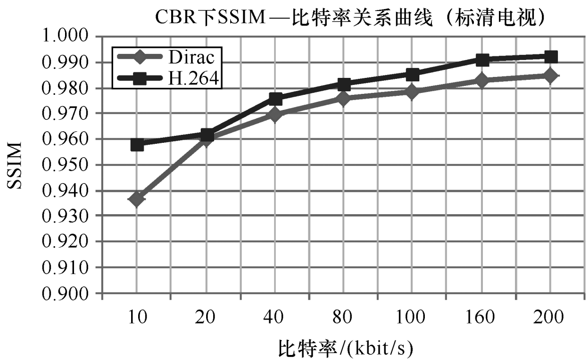


图 7. 18　针对标清电视序列“Susie”Dirac 和 H. 264 的 SSIM 对比［D15］

7. 5. 3　PSNR 测试

H. 264 能实现比 Dirac 高得多的 PSNR（大约 3 ~ 4dB），如图 7. 19 ~ 图 7. 21 所示。

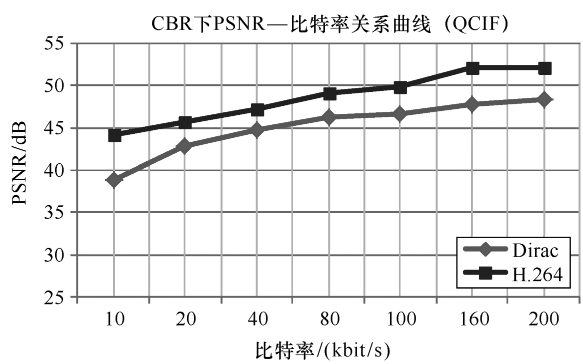


图 7. 19　针对 QCIF 序列“Miss- America”Dirac 和 H. 264 的 PSNR 对比［D15］

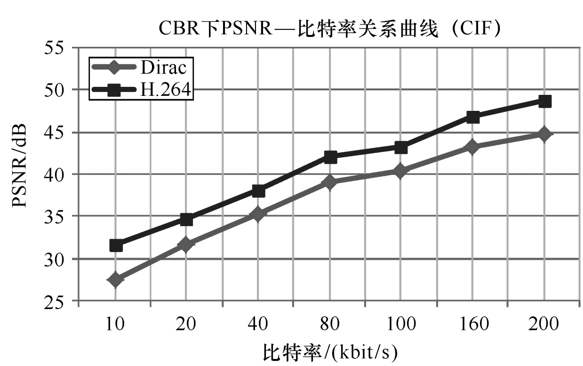


图 7. 20　针对 CIF 序列“Stefan”Dirac 和 H. 264 的 PSNR 对比［D15］

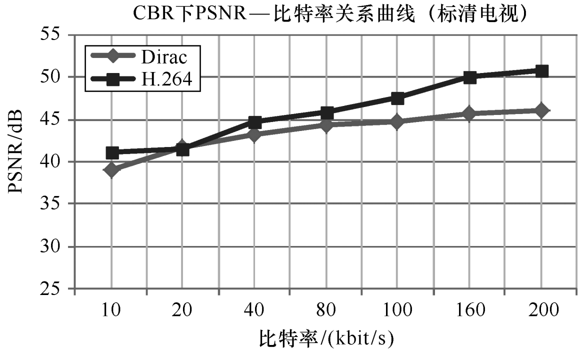


图 7.21　针对标清电视序列“Susie” Dirac 和 H. 264 的 PSNR 对比［D15］

表 7.1 ~ 表 7.3 以及图 7.22 ~ 图 7.24 分别给出了 Dirac 和 H. 264/MPEG-4 第 10 部分在 10KB/s 到 200KB/s 的固定比特率（CBR）上针对 QCIF、CIF 和标清序列的性能比较。

表 7.1　针对 QCIF 序列在 CBR 下 Dirac 和 H. 264 的性能比较

	Dirac				H. 264			
CBR（KB/s）	大小*（KB）	压缩比	PSNR 分贝值（Y）	SSIM	大小*（KB）	压缩比	PSNR 分贝值（Y）	SSIM
10	59	95	38. 913	0. 966	63	90	44. 162	0. 983
20	120	46	42. 911	0. 981	123	45	45. 729	0. 987
40	247	23	44. 648	0. 986	243	23	47. 257	0. 989
80	477	12	46. 180	0. 988	481	12	49. 054	0. 992
100	594	9	46. 640	0. 989	601	9	49. 826	0. 993
160	949	6	47. 717	0. 991	911	6	52. 077	0. 995
200	1186	5	48. 420	0. 992	912	6	52. 077	

* 表示压缩后所有 150 帧的编码文件大小。

表 7.2　针对 CIF 序列在 CBR 下 Dirac 和 H. 264 的性能比较

	Dirac				H. 264			
CBR（KB/s）	大小*（KB）	压缩比	PSNR 分贝值（Y）	SSIM	大小*（KB）	压缩比	PSNR 分贝值（Y）	SSIM
10	146	92	27. 468	0. 896	142	94	31. 617	0. 955
20	185	47	31. 613	0. 951	282	48	34. 650	0. 974
40	559	24	35. 296	0. 975	559	24	38. 055	0. 984
80	1，114	12	39. 012	0. 986	1，112	12	42. 134	0. 991
100	1，386	10	40. 343	0. 998	1，389	10	43. 134	0. 992
160	2，216	6	43. 273	0. 992	2，199	6	46. 840	0. 995
200	2，757	5	44. 684	0. 994	2，731	5	48. 729	0. 997

* 表示压缩后所有 90 帧的编码文件大小。

表 7.3　针对标清序列在 CBR 下 Dirac 和 H. 264 的性能比较［D15］

	Dirac				H. 264			
CBR（KB/s）	大小*（KB）	压缩比	PSNR 分贝值（Y）	SSIM	大小*（KB）	压缩比	PSNR 分贝值（Y）	SSIM
10	180	94	39. 055	0. 937	178	95	41. 028	0. 958
20	388	44	41. 729	0. 960	361	47	41. 530	0. 962
40	751	22	43. 22	0. 970	701	24	44. 814	0. 976
80	1，470	11	44. 276	0. 976	1，405	12	45. 871	0. 981
100	1，822	9	44. 676	0. 978	1，694	10	47. 491	0. 986
160	2，849	6	45. 589	0. 983	2，562	7	50. 016	0. 991
200	3，539	5	45. 988	0. 985	2，953	6	50. 819	0. 993

* 表示压缩后所有 25 帧的编码文件大小。

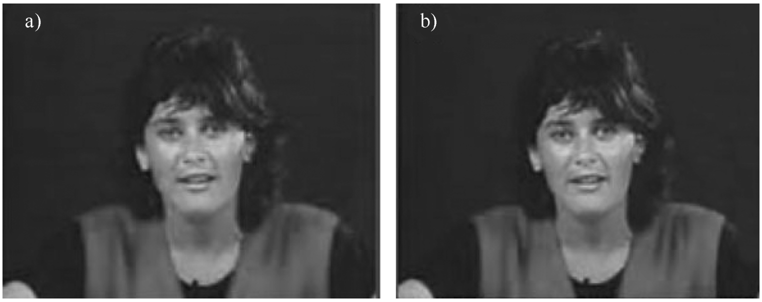


图 7.22　在 CBR = 10KB/s 时，Dirac 和 H. 264 的比较，QCIF 序列［D15］

a）Dirac 输出序列，CBR = 10KB/s PSNR：38. 913dB，比特率：9. 955KB/s［D25］

b）H. 264 输出序列，CBR = 10KB/s PSNR：44. 162dB，比特率：10. 603KB/s

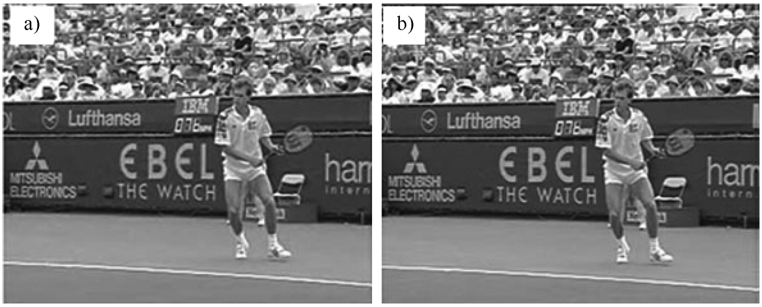


图 7.23　CBR = 100KB/s 时 Dirac 和 H. 264 的比较，CIF

a）Dirac 输出序列，CBR = 100KB/s PSNR：40. 343dB，比特率：101. 331KB/s［D15］

b）H. 264 输出序列，CBR = 100KB/s PSNR：44. 162dB，比特率：101. 546KB/s



图 7.24　CBR = 100KB/s 时 Dirac 和 H.264 的比较，标清电视［D15］

a）Dirac 输出序列，CBR = 100KB/s PSNR：44.676dB，比特率：103.606KB/s［D15］

b）H.264 输出序列，CBR = 100KB/s PSNR：47.491dB，比特率：96.324KB/s［D15］

7.6　结论

总体来看，Dirac 编译码器是非常有希望的。按照 BBC 研发部门［D1］［D25］，开发 Dirac 着眼于优化特别是压缩比和视觉质量方面的性能。它的简单性提供了非常有益的稳健性和压缩的快速性，因而 Dirac 在很大程度上已经实现了它的目标［D5］。

Dirac 是一种没有充分发展的编译码器，但这样一种早期的参考编译码器能产生与 H.264 可比的良好结果，这是值得称扬的。SSIM 表明，H.264 在质量方面具有略高的改善。编译码器的选择会取决于终端用户的应用，而应用会决定许可费的巨大代价是否抵得上质量的改善（如图 H.264/MPEG-4 第 10 部分的情况）［D5］。在低比特率下，Dirac 和 H.264 两者都保持接近固定的质量，这对于某些应用如流视频是有益的。

综上所述，Dirac 是一种及其简单却稳健的编译码器，并具有潜力实现非常接近于 H.264 的压缩效果，它的复杂度低且无需使用许可费用。然而，与这些编译码器的实现相比，H.264 肯定是获胜了。

7.7　未来的研究

Dirac 编译码器的实现瞄准从网络视频到超高清视频的高质量视频压缩。然而，该标准仅仅定义了一个视频编译码器而没有提到任何有关音频压缩的问题。因此，为了给终端用户提供有意义的视频传送，有必要将一个音频流与视频相关联在一起。

通过集成音频编译码器诸如 MPEG 第 2 层（MP2）［O34］或者 AAC［O16］，Dirac 视频编译码器可被进一步改善。MP2 是免使用许可费的，可用于高质量音频且在较高比特率下具有类似于 MP3 的性能［O35］。BBC 的 Dirac 研究组还建议 Vorbis［O37］音频编译码器和 Xiph. Org 基金会的开发 FLAC（免费的无损音频编译码器）［O36］作为高质量音频格式，这些格式可在适用于 Dirac 视频编译码器的使用许可免费条款下获得。Urs 已经用 AAC 音频实现了这一建议［D16］。

因此，通过复用视频和音频编码的传输比特流然后在接收端解复用这些流，有可能实现信号的传输。在此之后，可以在回放期间对音频和视频实现同步化以便于它能适用于各种应用。

7.8　总结

Dirac 不同于其他编译码器（AVS China，H.264/AVC，HEVC，VC-1 etc）的地方在于它基于小波变换而不是 DCT/整数 DCT。本章描述了其编码器，然后是与 H.264/AVC（第 4 章）的对比。最后一章（第 8 章）描述了 VC-1 视频编码（基于微软的 Windows Media Video 9，被 SMPTE 采用）。我们可以看到 AVS China、H.264/AVC 和 VC-1 之间的相似性。SMPTE 已经采用 Dirac 的帧内编码，就像 VC-2 一样。

7.9　专题研究

P.7.1　Ravi［D14］针对 QCIF、CIF 和标清电视序列实现了 Dirac 并且与 H.264［H23，H3，H29，H25，H13，H31，H20］进行了对比。参见图 7.13～图 7.15（压缩比与比特率的关系）。请用 Dirac［D21］［D20］和 JM［H30］软件对这些视频实现这些比较。还要进行其他比较，包括 SSIM 与比特率的关系（见图 7.16～图 7.18）和 PSNR 与比特率的关系（图 7.19～

图 7.21)。再把这些比较推广到 SSIM 的情况 [Q13]。

P.7.2 Urs [D16] 已经实现了对 Dirac 视频和 AAC 音频的编码和解码，同时保证了唇同步（原文：lip sync——译者注）。请用不同的视频序列实现它。
提示：要访问 Ravi [D14] 和 Urs [D16] 的毕业论文，请到 www-ee.uta.edu/dip，点击 courses，然后点击 EE5359。向下滚动窗口并查看列表：list of theses/projects。毕业论文、专题、议案、ppt 等可以下载。

P.7.3 用 Vorbis [O37] 音频代替 AAC 音频并重复 P.7.2。

P.7.4 用 FLAC（免费且无损的音频编译码器）[O36] 代替 AAC 音频并重复 P.7.2。

P.7.5 Dirac Pro [D23] 在 Dirac 内仅仅基于帧内编码。SMPTE 采纳它作为 VC-2。Shreyanka Subbararyappa 已经实现了这个专题，“Comparative study of H.264 intra frame coding，JPEG，JPEG-LS，JPEG 2000 and JPEG XR”。该专题的议案/报告以及它们的 ppt 都可以获取到。Radhika Veerla 实现了类似的专题，题名为“Advanced image coding”。请实现这两个专题并与 Dirac Pro 就若干测试序列（QCIF、CIF、SDTV、HDTV 格式）在各种比特率下进行对比。议案、报告和 ppt 可供访问。
提示：为了访问这两个专题，请参见 P.7.2。向下滚动窗口并查看“previous projects”，Shreyanka Subbarayappa：Fall 2010，Radhika Veerla：Summer 2008.

第 8 章　VC-1 视频编码

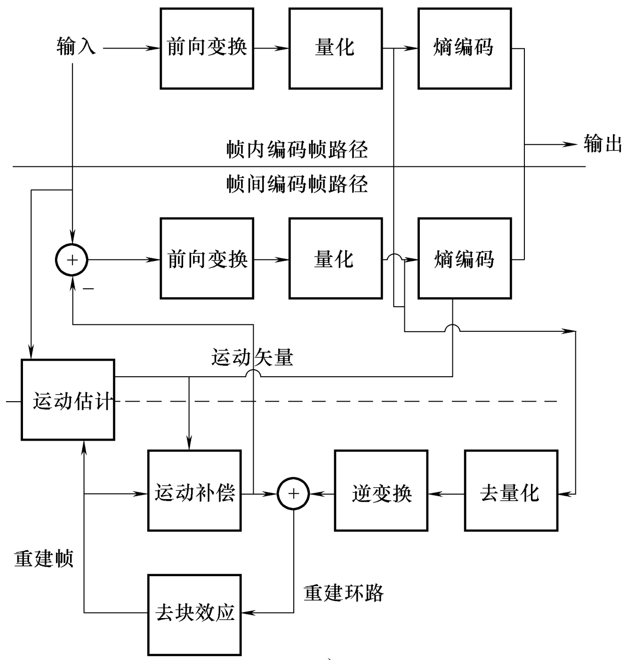
摘要：与其他编译码器一样，本章描述了 VC-1 编码器/译码器的细节。VC-1 是一个基于微软开发的 WMV9 的 SMPTE 标准。本章还概述了 H. 264/AVC 到 VC-1 的转码器。SMPTE 只采纳了 Dirac（第 7 章）的帧内编码作为 VC-2。

关键词：VC-1　SMPTE　微软　档次　Z 型模式　重叠变换　自适应变长编码　转码　VC-1 的传输　VC-2 视频压缩　H. 264 到 VC-1 的转码器

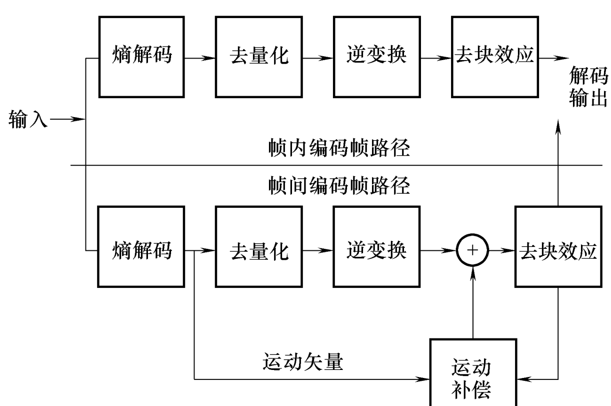
8.1　VC-1 的结构

VC-1 是 SMPTE 421M［C7］视频编译码标准的正式名称，它最初是由微软公司开发的，由 SMPTE 于 2006 年 4 月 3 日发布。现在，它是蓝光光盘和视窗媒体视频流格式 9（WMV9，Windows Media Video 9）的支持标准。VC-1 是传统上基于 DCT 的视频编译码器设计方案的一种演化，这种设计在 H. 261［S4］，H. 263［S6］，MPEG-1［S2］，MPEG-2［S3］和 MPEG-4［S10］也能发现。它被广泛地作为最新的 ITU-T 和 MPEG 视频编译码标准 H. 264/MPEG-4 AVC［S10］的替代品。VC-1 包括隔行视频序列的编码工具以及渐进编码的工具。开发和标准化 VC-1 的主要目的是支持隔行内容的压缩，而不必首先转换为渐进格式，从而使得它对于广播和视频工业界的专业人士更具吸引力。

VC-1 编译码器被设计来实现最新的压缩质量，比特率范围可以从很低到很高。该编译码器能很容易地为高清视频在 6 ~ 30Mbit/s 上处理 1920 × 1080 像素的分辨率。VC-1 能够处理更高的分辨率如数字电影的 2048 × 1536 像素，和最大为 135Mbit/s 的比特率。一个甚低比特率视频的例子是面向调制解调器应用的在 10kbit/s 上的 160 × 120 像素分辨率。VC-1 的这个基本功能涉及一个基于块的运动补偿和空间变换方案，类似于用在其他视频压缩标准如 MPEG-1 和 H. 261 的方案，如图 8. 1 所示。然而，VC-1 包括若干创新和优化，使得它与基本的压缩方案有所不同，从而产生了卓越的质量和效率。VC-1 先进档次与传输无关。这一点对于设备厂商和内容业务提供了更大的灵活性。



a)



b)

图 8.1　a）VC-1 编码器框图。b）VC-1 解码器框图［C2］（2004 Elsevier）

8.2　整数变换编码

8.2.1　逆变换

VC-1 使用整数编码将样值块转化到变换域，这样可以避免逆变换失配问题。整数变换是离散余弦变换（DCT，Discrete Cosine Transform）的一种变体，DCT 已

被很多国际视频编码标准广泛采纳。VC-1 和其他编译码器的主要区别是它基于可变大小的变换进行工作。变换块大小可以是 8×8 或者三种亚块大小（8×4，4×8，4×4）的任何一个，取决于块的特性。假定所有的变换方法都能适应块特性从而最小化变换误差。方程（8.1）和方程（8.2）分别给出 8×8 和 4×4 的逆变换。

$$T_8=\begin{bmatrix}12&12&12&12&12&12&12&12\\16&15&9&4&-4&-9&-15&-16\\16&6&-6&-16&-16&-6&6&16\\15&-4&-16&-9&9&16&4&-15\\12&-12&-12&12&12&-12&-12&12\\9&-16&4&15&-15&-4&16&-9\\6&-16&16&-6&-6&16&-16&6\\4&-9&15&-16&16&-15&9&-4\end{bmatrix} \tag{8.1}$$

$$T_4=\begin{bmatrix}17&17&17&17\\22&10&-10&-22\\17&-17&-17&17\\10&-22&22&-10\end{bmatrix} \tag{8.2}$$

对于帧内编码的宏块总是使用 8×8 变换，而对于帧间编码的宏块，每个 8×8 块可能用一个 8×8，两个 4×8，两个 8×4，或四个 4×4 进行变换。块大小可变的变换可在序列级、帧级、宏块级、或块级上给出指示信息。

逆变换过程的第一步是计算直接数据块 $E_{M\times N}$。

$$E_{M\times N}=(D_{M\times N}T_M+4)\gg 3 \tag{8.3}$$

式中，$D_{M\times N}$是输入变换域数据块；T_M 是整数 DCT 矩阵；分别如式（8.1）和式（8.2）所示。操作符‘≫’表示在矩阵上执行逐项的算术右移。变换大小参数是 4 或 8，即 M，$N\in\{4, 8\}$。这样，在 VC-1 中就有可能存在 8×8，4×8，8×4 和 4×4 的变换。逆量化后的变换系数值不会超过带符号 12bit 的值范围（即 $-2048\leq E_{M\times N}$的元素≤ 2047），而直接矩阵 $E_{M\times N}$的值范围是 13-比特（即 $-4096\leq E_{M\times N}$的元素≤ 4095）。

第二步是计算逆变换数据块 $R_{M\times N}$

$$R_{M\times N}=(T_N E_{M\times N}+C_N 1_M+64)\gg 7 \tag{8.4}$$

式中，$C_4=(0\ 0\ 0\ 0)'$和 $C_8=(0\ 0\ 0\ 0\ 1\ 1\ 1\ 1)'$。$1_M$ 定义为一个长度为 M 元素为 1 的行向量。逆变换值的范围在 10bit 以内（即 $-512\leq E_{M\times N}$的元素≤ 511）。

8.2.2　前向变换

因为前向变换没有考虑进行快速实现，故它可以缩放的整数算法或用浮点或其他表示来实现。实现前向变换的矩阵乘法表示纯粹是一种解析表示。四种不同块大小（4×4，8×4，4×8 和 8×8）的前向变换系数可以如下计算

$$\hat{D}_{4\times4} = (T_4\ D\ \hat{T}_4) \circ N_{4\times4}$$
$$\hat{D}_{8\times4} = (T_8\ D\ \hat{T}_4) \circ N_{8\times4}$$
$$\hat{D}_{4\times8} = (T_4\ D\ \hat{T}_8) \circ N_{4\times8}$$
$$\hat{D}_{8\times8} = (T_8\ D\ \hat{T}_8) \circ N_{8\times8} \tag{8.5}$$

式中，操作符○是一种逐分量乘法，标准化矩阵 $N_{M\times N}$的元素由以下给出：

$$N_{ij} = c_j c_i' \tag{8.6}$$

注意，i 和 j 表示标准化矩阵的行和列号，列向量 c 是

$$c_4 = \begin{pmatrix} \frac{8}{289} & \frac{8}{292} & \frac{8}{289} & \frac{8}{292} \end{pmatrix}'$$

$$c_8 = \begin{pmatrix} \frac{8}{288} & \frac{8}{289} & \frac{8}{292} & \frac{8}{289} & \frac{8}{288} & \frac{8}{289} & \frac{8}{292} & \frac{8}{289} \end{pmatrix}' \tag{8.7}$$

每个基函数不同的标准化因数表明，VC-1 定义的变换仅仅是正交的，而不是标准正交的。因此有关正交变换的传统论述不能成立。

1. Z 型扫描

将样值数据变换到频域之后，变换系数以 Z 字形样式进行重新排序以使后续的编码方法更为有效。VC-1 有 13 中不同的 Z 字形扫描样式，取决于不同的编码参数，诸如块大小、隔行还是渐进、预测模式和帧内或帧间模式。

有三种不同的 Z 字形扫描样式用于帧内块，如图 8.2 所示：帧内普通扫描、帧内水平扫描、和帧内垂直扫描。对于帧间编码块，定义了四种取决于块大小的不同的扫描样式，如图 8.3 所示：帧间 8×8 扫描、帧间 8×4 扫描、帧间 4×8、和帧间 4×4 扫描。还定义了其他六种不同的扫描样式：渐进 8×4 扫描、渐进 4×8 扫描、隔行 8×8 扫描、隔行 8×4 扫描、隔行 4×8 扫描、和隔行 4×4 扫描。

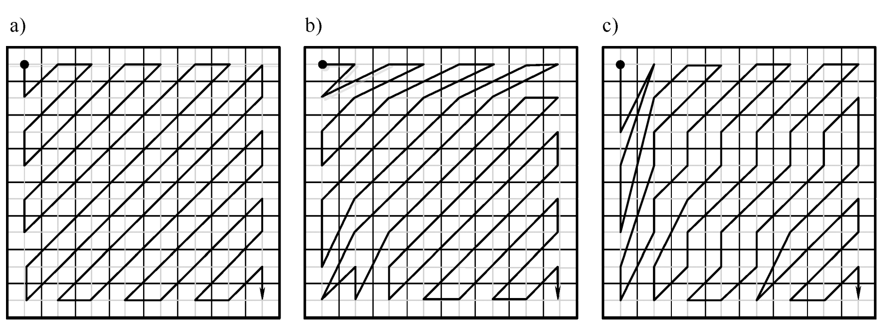


图 8.2　VC-1 中帧内 Z 字形扫描

a）标准的　b）水平的　c）垂直的［B14］（2008　Springer）

2. 量化

量化是一个实际压缩的步骤，在按照预定顺序扫描之后它忽略了不重要的信息。有两种量化方法，取决于量化步长的大小：均匀和非均匀。量化过程基于某种

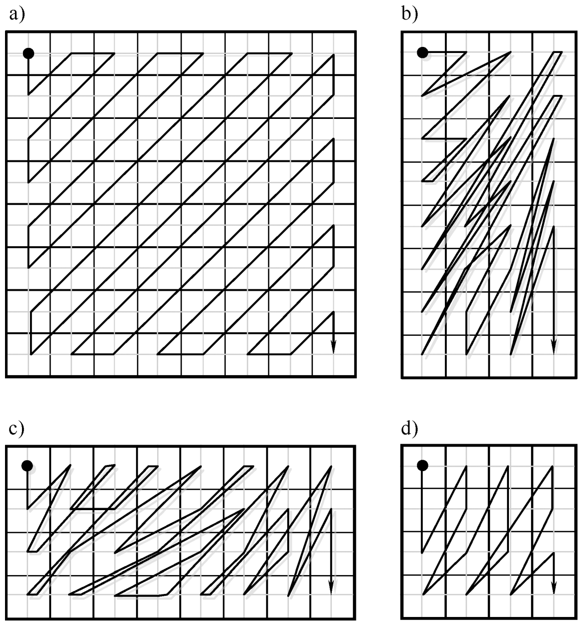


图 8.3　VC-1 帧间编码的 Z 字形扫描（2008 Springer）

准则诸如量化器失真最小化，用数量有限的级表示数据。由于输出值表示了步长间隔范围里的很多输入值，所以量化是一个固有的有损过程。均匀量化器意味着输入和输出步长是均匀分布的，而非均匀量化器则包含了不均匀的分布。

VC-1 编码器将整个帧的量化器等级指数（5bit 的 PQINDEX）发送到解码器。它在所有类型的帧都存在。实际的量化器等级（PQUANT）以均匀或非均匀的方式从 PQINDEX 转化得到。宏块量化器（MQUANT）可基于宏块进行修改。那些位于帧边缘的宏块以另一个量化步长（ALTPQUANT）来量化，而其他所有的宏块以 PQUANT 进行量化。每个单一的宏块可能会以不同方式进行量化。在此情况下，就会指出每个宏块是否可以从两个量化步骤（PQUANT 或 ALTPQUANT）中选出，或者是否每个宏块可以用任何步长进行任意的量化。

8.3　运动估计/补偿

运动估计（ME）是视频编码中计算强度最大的部分，因为它试图逐像素或逐块地在参考帧中找到与当前帧内块的匹配块。视频的运动补偿（MC）通过从当前块减去匹配块（由运动矢量指明）得以实现。块匹配得越近，两个块之间的差别就越小。

为了运动估计，一个 16×16 的亮度宏块可被分成四个 8×8 的子块。当覆盖区域内的（如物体边界上的）纹理运动不够均匀时，8×8 子块的运动补偿相当有效。

例如，当一个宏块落入背景时，物体边界宏块的四个 8×8 块之一可能以不同的方向移动，而另外三块则处于前景中。在此情况下，运动是不均匀的，较小区域的运动补偿能提高更好的压缩性能。这一策略已被以前的标准如 MPEG-4 第 2 部分所采纳［S9］。VC-1 向前更进一步。宏块里的任一个 8×8 块都可以帧内模式编码，而其他块以帧间/运动补偿模式进行编码。

VC-1 使用半像素和四分之一像素的帧间运动补偿，伴随着或者是双线性（像 H.264 那样）或双三次内插。更多细节在 8.4.6 节中解释。VC-1 提出如下四种 MC 方法：

1）16×16 块-单运动矢量-半像素-双线性；

2）16×16 块-单运动矢量-半像素-双三次；

3）16×16 块-单运动矢量-1/4 像素-双三次；

4）8×8 块-四运动矢量-1/4 像素-双三次。

注意，安排的顺序是复杂度和质量递增的顺序。任何 MC 方法的选择取决于应用场合。例如，移动手持设备计算能力较低，那么就不需要高图像质量的视频。在此情况下，16×16 块-单运动矢量-半像素-双线性内插的 MC 对于编码器就是一个好的选择。然而，对于高清 DVD，就需要 8×8 块-四运动矢量-1/4 像素-双三次内插的 MC 来实现卓越的视频质量。在需要 1/4 分辨率的时候，半像素内插是一个中间过程。

8.3.1 环路滤波器

粗糙的量化会引起块边界处的不连续，产生可见的块效应。不精确的运动补偿也会引起块边界处可见的瑕疵，叫作“飞蚊”效应。为了消除这些影响采纳了去块效应滤波器，用在重建帧作为后续帧的参考帧之前。编码器和解码器必须执行相同的滤波操作。

VC-1 有两种消除这些效应的方法——重叠变换（OLT，Overlapped Transform）和环路滤波器（ILF，In-loop Filtering）［V1］。OLT 通过平滑边界的不连续性对于涉及一个高质量块和一个低质量块的边界是有用的，而 ILF 通过增强图像质量对于那些涉及两个低质量块或两个高质量块的边界是有用的。OLT 是一个独特的方法，它基于一对精确定义的预/后处理。这个想法是定义这样的前向和逆向操作并连续执行的话就能完美地恢复原始数据。前向变换跨越相邻块边界交换信息。一般情况下，在一个块边沿上，一个块有相对好的边沿细节而另一个则不然。解码器需要一种逆向操作将边沿数据交换回来以降低块效应。通过这种方法，高质量和低质量边沿对互相扩散，从而提高了视觉质量。

在 P 帧的环路滤波中，一个块及其相邻块之间的边界是不滤波的，如果两个块具有相同的运动矢量且如果两个块都没有残差。在量化或运动补偿引入不连续性不太可能发生地方，这防止了块边界的过度平滑。这一限制也降低了 P 帧 ILF 过程的复杂度。围绕块边沿发生突变的时候，斑驳的图案中高频成分较多。考虑到原始

质量的图像中也可能包含高频成分，这一过程围绕 I 和 P 参考帧的块边沿使用了一个相对简单的非线性低通滤波器。这样，滤波的结果只会影响到以滤波帧为参考的那些后续帧。

8.3.2　复杂度

VC-1 具有比 H.264 更低的计算复杂度，见表 8.1 所示。比较编码特征时，在给定的比特率下 H.264 给出了比 VC-1 略好的质量，但它也具有较高的复杂度，这是由于在帧内预测、运动补偿、和宏块编码模式方面的高度灵活性导致的。例如，与算术编码相比，使用 VLC 表不会给出最好的性能，但它的复杂度也低一些。因此，VC-1 所进行的选择代表了质量与复杂度之间的平衡。

表 8.1　VC-1 和 H.264 的特征比较

特征	VC-1 简单档次	H.264 基本档次
帧类型	I、P	I、P
变换块大小	4×4，4×8，8×4，8×8	4×4
变换	整数 DCT	整数 DCT
帧内预测	频域直流和交流系数预测	4×4，16×16 空域，IPCM
运动补偿块大小	16×16，8×8	16×16，16×8，8×16，8×8，8×4，4×8，4×4
宏块模式总数	3	7 种帧间模式 +（9+4）种帧内模式 8×4
运动矢量分辨率	1/4 像素	1/4 像素
去块效应	环路滤波器，重叠变换	环路滤波器
参考帧	单帧	单帧、多帧
熵编码	自适应 VLC	CAVLC

8.3.3　档次和级别

视频编译码器定义了一个工具和算法的子集，诸如双预测帧、起始码、亮度补偿等，以及档次内的级别，它对定义特定档次的参数又作了约束。VC-1 定义了如图 8.4 所示的三种档次：

1）简单档次（SP，Simple Profile）服务于低比特率互联网流媒体和低复杂度应用诸如移动通信或掌上电脑（原文：personal digital assistant——译者注）中的媒体回放。SP 包括两个级别——低级和中级，见表 8.2。

2）主档次（MP，Main Profile）服务于高速率互联网应用，诸如通过 IP 传送的流视频/音频，或者 IP 网络上的电视/视频点播。MP 包括三个级别——低级、中级和高级。

3）高级档次（AP，Advance Profile）服务于广播应用，诸如数字电视、PC 回放的高清 DVD，或高清电视。它是唯一支持隔行内容的档次。此外，这个档次包含所需的语法元素来传输视频比特流。AP 包括五个级别——L0 到 L4。

表 8.2　档次和级别的限制

档次@级别	MB/s	MB/f	例　　子	P	I	Rmax	Bmax	MV([H]×[V])
SP@LL	1,485	99	QCIF，176×144，15Hz			96	20	[−64，63¾]×[−32，31¾]
SP@ML	7,200	396	CIF 352×288，15Hz			384	77	[−64，63¾]×[−32，31¾]
			QVGA，320×240，24Hz					
MP@LL	11,880	396	QVGA，320×240，24Hz	x		2,000	306	[−128，127¾]×[−64，63¾]
			CIF 352×288，30Hz					
MP@ML	40,500	1,620	480p，720×480，30Hz	x		10,000	611	[−512，511¾]×[−128，127¾]
			576p，720×576，25Hz					
MP@HL	245,760	8,192	1,080p，1,920×1,080，25Hz	x		20,000	2,442	[−1,024，1,023¾]×[−256，255¾]
			1,080p，1,920×1,080，30Hz					
AP@L0	11,880	396	CIF，352×288，25Hz，	x		2,000	250	[−128，127¾] 9×[−64，63¾]
			CIF，352×288，30Hz					
			SIF，352×240，30Hz					
AP@L1	48,600	1,620	480i-SD，704×480，30Hz	x	x	10,000	1,250	[−512，511¾]×[−128，127¾]
			576i-SD，720×576，25Hz					
AP@L2	110,400	3,680	480p，704×480，60Hz	x	x	20,000	2,500	[−512，511¾] 9×[−128，127¾]
			720p，1,280×720，25Hz					
			720p，1,280×720，30Hz					
AP@L3	245,760	8,192	1,080i，1,920×1,080，25Hz	x	x	45,000	5,500	[−1,024，1,023¾]×[−256，255¾]
			1,080i，19209×1,080，30Hz					
			1,080p，1,9209×1，080，25Hz					
			1,080p，1,920×1，080，30Hz					
			720p，1.280×720，50Hz					
			720p，1280×720，60Hz					
			2,048×1,024，30Hz					
AP@L4	491,520	16,384	1,080p，1,920×1,080，50Hz	x	x	135,000	16,500	[−1.024，1.023¾]×[−256，255¾]
			1,080p，1,9209×1,080，60Hz					
			2,048×1,536，24Hz					
			2,048×2,048，30Hz					

MB/s：每秒最大宏块数，MB/f：帧内最大宏块数，Rmax：以 kbit/s 为单位的最大比特率，Bmax：以 16,384 比特为单位的最大缓冲空间大小，MV：([H]×[V])：运动矢量范围（[水平]×[垂直]），P：渐进的，I：隔行的，respectively [C7]。(2006 SMPTE)

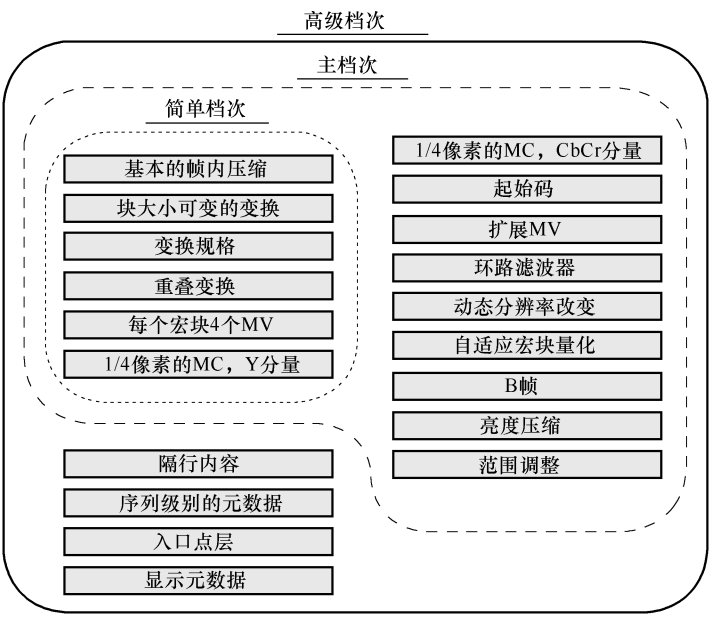


图 8.4　VC-1 档次和级别中的编码成分

VC-1 的语法包括多个分级层次：序列、入口点、帧、场、分片、宏块、和块见表 8.3。在简单和主档次中，一个序列包括一系列一个或多个编码帧，这些编码帧分解成多个宏块，每个宏块包括四个亮度块和两个色差块，如图 8.5 所示。在高级档次中，一个序列包括一系列一个或多个入口点片段，每个入口点片段包括一系列一个或更多帧，其中每个入口点片段的第一帧提供了随机访问功能。高级档次的一个分片包括一个或多个相邻行的宏块。在隔行高级档次中，给出了两个场的对应层。

表 8.3　VC-1 档次中的分级比特流层次［B14］

	档次		
	简单与主档次	渐进的高级档次	隔行的高级档次
比特流层	1. 序列 2. 帧 3. 宏块 4. 块	1. 序列 2. 入口点 3. 帧 4. 分片 5. 宏块 6. 块	1. 序列 2. 入口点 3. 帧 4. 场 5. 分片 6. 宏块 7. 块

2008Springer

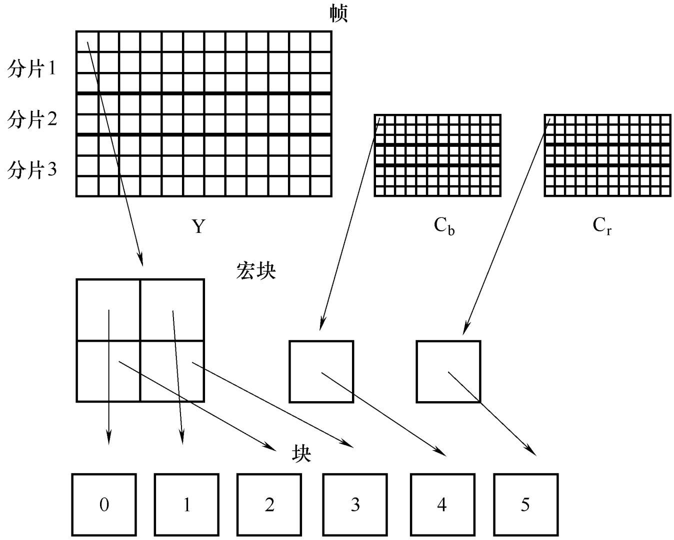


图 8.5 给出 VC-1 的帧、分片、宏块和块的编码层次［C7］（2006 SMPTE）

8.4 简单档次

8.4.1 比特流结构

VC-1 解码器用来自编码器的元数据初始化它的解码过程。表 8.4 为简单档次和主档次定义的序列层数据结构，以码值 0xC5 开头，后面是若干编码帧和相关参数。

STRUCT_C 包括如下的若干参数。PROFILE（4bit）指出简单档次、主档次还是高级档次。后处理量化帧率指示器（FRMRTQ_POSTPROC，3bit）给出后处理的帧率。如果它是 7，那么帧率大约是 30 帧/s。输入的其他值通过方程（2 + FRMRTQ_POSTPROC ×4）导出帧率。比特率由 BITRTQ_POSTPROC（5bit）进行类似定义。LOOPFILTER 是 1bit 的逻辑量，对于简单档次应该是 0 值（无环路滤波）。通过使用 1bit 的 MULTIRES，任何一帧都可以低于规定帧分辨率（垂直和水平大小）进行编码。多分辨率的实际大小可以按照帧级上的 RESPIC 来定义。FASEUVMC（1bit）指明是否要使用亚像素内插和色差运动矢量取整。如果它设为 1，1/4 精度的色差运动矢量会取整到最近的半像素或全像素的位置。EXTENDED_MV（1bit）指明是否使用扩展运动矢量，对于简单档次总是设为 0（没有扩展的运动矢量）。可以使用不同的量化器（DQUANT，2bit），如果在主档次中该标志等于 1 或 2。对于简单档次，DQUANT =0；对于整幅帧只用一个量化步长。可变大小的变换标志（VSTRANSFORM，1bit）用来指出变换块的大小可以变

为 8×4，4×8，或 4×4。如果 OVERLAP（1bit）设为 1，那么使用重叠变换。同步标记（SYNCMARKER，1bit）在简单档次中设置为零。如果 SYNCMARKER == 1，那么同步标志就会出现在比特流中。在简单档次中范围减少标志（RANGERED，1bit）会设为 0，伴随它的有帧层里的 RANGEREDFRM，该参数指出在保证原始重建帧用于未来运动补偿的同时，当前解码帧应该按比例放大。MAXBFRAMES（3bit）指出 I 或 P 帧之间 B 帧的个数。QUANTIZER（2bit）指示用于序列的量化器，可以是隐含的（00b）或显式的（10b），和均匀的（10b）或非均匀的（11b）。帧内插标志（FINTERPFLAG，1bit）偕同帧首部中的 INTERPFRM，指示启用帧内插。

表 8.4　简单档次和主档次的序列层数据结构 [C7]

31 ~ 24（1B）	23 ~ 0（3B）
0xC5（8bit）	NUMFRAMES（24bit）
0x00000004（4B）	
STRUCT_C 包括 PROFILE，FRMRTQ_POSTPROC，BITRTQ_POSTPROC，LOOPFILTER，MULTIRES，FASTUVMC，EXTENDED_MV，DQUANT，VSTRANSFORM，OVERLAP，SYNCMARKER，RANGERED，MAXBFRAMES，QUANTIZER，FINTERFLAG（4B）	
STRUCT_A 包括 VERT_SIZE（4B）	
STRUCT_A 包括 HORIZ_SIZE（4B）	
0x0000000C（4B）	
STRUCT_B 包括 LEVEL（3bit），CBR（1bit），RES1（4bit），HRD_BUFFER（24bit）	
STRUCT_B 包括 HRD_RATE（4B）	
STRUCT_B 包括 FRAMERATE（4B）	

STRUCT_A 包括垂直帧大小（VERT_SIZE，32bit）和水平帧大小（HORIZ_SIZE，32bit）。简单档次和主档次中的 STRUCT_B 包括六个参数：指示编码级别的 LEVEL（3bit），指示恒定比特率模型是否被用的 CBR（1bit），以 ms 为单位指示假设参考解码器缓冲大小的 HRD_BUFFER（24bit），以 bit/s 为单位指示传输速率峰值的 HRD_RATE（32bit），和指示编码视频段的取整帧率的 FRAMERATE（32bit）。

8.4.2　基本档次的帧内压缩

压缩过程首先从执行变换编码降低空间冗余开始，然后进行基于块的预测编码以降低时间冗余。对一个帧内编码块来说，一帧被分解为多个块然后进行 8×8 的整数变换得到变换系数，如图 8.6 所示。注意到，帧内编码块仅仅利用来自自身或当前帧的信息；而帧间编码块用到自身和其他时间上出现的块和帧的信息。量化用于变换系数，产生量化过的 1 个直流系数和 63 个交流系数。

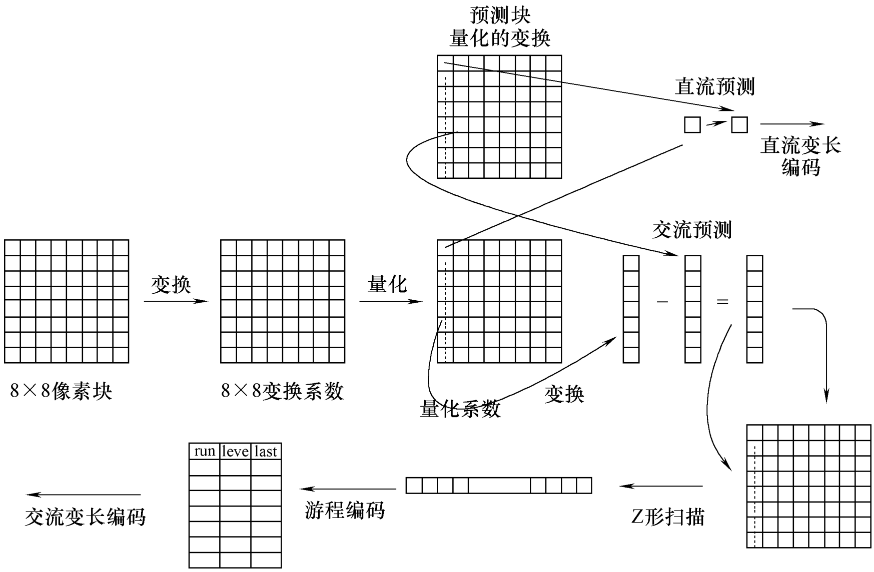


图 8.6 帧内编码块的变换编码［C7］（2006 SMPTE）

直流系数通过给量化直流系数乘以量化步长得以重建。

$$DC_{coef} = DC_Q \times DCStepSize \quad (8.8)$$

DCStepSize 值从宏块量化值（MQUANT）中得出如下：

如果 MQUANT 等于 1 或 2，DCStepSize = 2 × MQUAN。

如果 MQUANT 等于 3 或 4，DCStepSize = 8。

如果 MQUANT 大于或等于 5，DCStepSize = MQUAN/2 + 6。

标准定义了均匀和非均匀量化器交流系数的逆量化。如果均匀量化器由参数 QUANTIZER 指示，

$$AC_{coef} = AC_Q \times (2 \times MQUANT + HALFQP) \quad (8.9)$$

如果使用了非均匀量化器，

$$AC_{coef} = AC_Q \times (2 \times MQUANT + HALFQP) + sign(AC_Q) \times MQUANT \quad (8.10)$$

式中，AC_Q 表示量化的交流系数；HALFQP 是出现在各类型帧中的 1bit 的语法元素。如果 HALFQP == 1，当进行基于 PQUANT 进行解码时，帧量化器步长等于 PQUANT + 0.5。当进行基于 VOPDQUANT 的解码时，它是零。如果 PQINDEX 小于或等于 8，它出现在各种类型的帧里。

8.4.3 块大小可变的变换规定

通过令序列层中的语法元素 VSTRANSFORM = 1，启用块大小可变的变换编码。这一元素用于所有的三种档次中。如果 VSTRANSFORM == 0，8 × 8 的变换用于所

有的块。如果 VSTRANSFORM = = 1，宏块级的变换类型标记（TTMBF，Macroblock-Level Transform Type Flag，1bit 的语法元素）也出现在 P 和 B 帧的首部，指明是否在帧或宏块级上启用变换类型编码。如果 TTMBF = = 1，那么帧级上的变换类型（TTFRM：frame-level transform type）语法元素就会出现在帧级的层里。变换的大小由表 8.5 的 2bit TTFRM 码来定义，它用于帧里的所有块。

表 8.5 TTFRM 码对应的可变变换的大小信息

TTFRM 码	变换类型
00b	8 ×8 变换
01b	8 ×4 变换
10b	4 ×8 变换
11b	4 ×4 变换

如果启用块大小可变的变换，那么 8 ×8 的残差块可能用一个 8 ×8 变换来变换，也可能水平划分然后进行两个 8 ×4 变换，或者垂直划分然后进行两个 4 ×8 变换，或者分成四部分进行四个 4 ×4 变换，如图 8.7 所示。

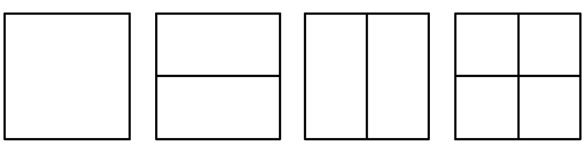


图 8.7 块大小可变变换的结构［C7］（2006 SMPTE）

如果 TTMBF = = 0，那么变换类型在整幅帧上都会改变，而且应该在宏块级或者块级上加以指示。宏块变换类型（TTMB，Macroblock Transform Type）在宏块层定义了变换类型，指示级别（宏块还是块），和子块模式。出现 TTMB 表明，至少在宏块中有一个块具有非零系数（即至少一个编码块）。如果指示级别规定了宏块模式，那么 TTMB 定义的变换类型就用于解码宏块中的所有块。如果指示级别是块模式，那么变换类型就用于解码宏块中的第一个编码块。其他块的变换类型在块级别上给出。如果变换类型是 8 ×4 或 4 ×8，那么子块模式就给出 8 ×4 或 4 ×8 子块的哪一个具有至少一个非零系数。表 8.6 中用示例总结了变换大小和相关指示，因为 TTMB VLC 取决于 PQUANT 的值。如果他小于或等于 4，就定义高比特率的 VLC。如果它大于 4 且小于或等于 12，就定义中等比特率的 VLC。如果它大于 12，就定义低比特率的 VLC。表 8.6 中的示例代码是中等比特率的。

表 8.6 定义块大小可变的变换的参数

	指示	启用	示例	意义	启用	
VSTRANSFORM ==0					8×8 变换	
VSTRANSFORM ==1	TTMBF ==0	TTMB	0010001b		8×4 变换	
			(VLC)		宏块模式	不出现 TTBLK
					上 8×4 编码块	
			1111b		4×8 变换	
			(VLC)		块模式	出现 TTBLK
					右 4×8 编码块	
		TTMBF ==1	TTFRM	00b	8×8 变换	
				01b	8×4 变换	
				10b	4×8 变换	
				11b	4×4 变换	

变换大小块级定义的 TTBLK 语法元素只出现在帧间编码的块中，且只在 TTMB 给出块模式的宏块中。每个宏块中的第一个帧间编码块没有对应的 TTBLK，因为变换类型和子块模式是用于第一个帧间编码块的。这样，对于第一个帧间编码块之后的每个帧间编码块都对应一个 TTBLK。还定义了 TTBLK 相应于低、中、高 PQUANT 值的三个不同的 VLC 表。这些只适用于 8×4 和 4×8 变换。对于 4×4 变换类型，出现 SUBBLKAT 来规定四个 4×4 子块的哪一个至少有一个非零系数。SUBBLKPAT 的三个 VLC 表也依据 PQUANT 的值来定义。

8.4.4 重叠变换

序列层的语法元素 OVERLAP 用于指示对于亮度和色差通道是否跨越两个相邻帧内编码块的边沿进行有条件的滤波操作（OVERLAP ==1）还是不操作（OVERLAP ==0）。该滤波操作被称为是重叠平滑。对于简单档次，仅当帧级量化步长 PQUANT 是 9 或更高时才用到 OLT。这是因为 OLT 只对具有失真边沿的块才起作用。

重叠变换（OLT）的基本思想是交换两个相邻块的边沿数据，两个块都具有原始质量，如图 8.8 所示。当两个这样的相邻块经过变换/量化和逆量化/逆变换，在某些情况下，在一个块中就可能引入比其他块更严重的量化误差和/或块效应。在解码器，两侧的边缘数据应该再次交换位置以恢复原始数据的位置。那么，一个好质量的块包含了差质量的边缘，而一个差质量的块包含了好质量的边缘。换言之，好质量和差质量块彼此融合。

当直接应用重叠变换平滑（见图 8.8）的概念时，由于边缘交换就可能引入高频分量。因此，把滤波操作定义为 OLT 而不是简单的数据置换。重叠平滑用于 8×

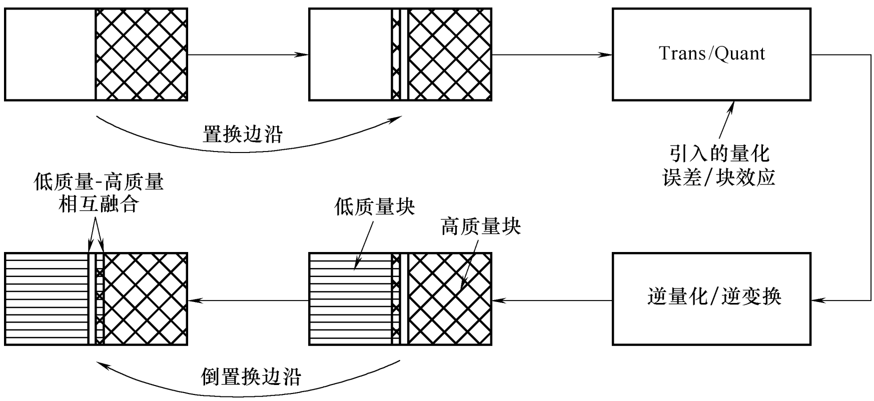


图 8.8　重叠变换平滑的概念［B14］（2008 Springer）

8 的帧内块边沿，边界两端的两个像素如图 8.9 所描述，给出在水平方向上平滑垂直边沿的四个像素（a0，a1，b1 和 b0），以及在垂直方向上平滑水平边沿的四个像素（p0，p1，q1，q0）。黑色圆圈标记在两个方向上被平滑的 2×2 的角落子块。首先对垂直边沿进行滤波，之后进行水平滤波。

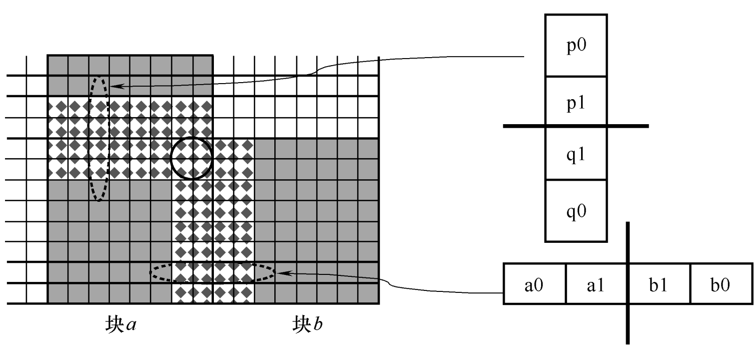


图 8.9　两块之间重叠平滑示例［C7］（2006 SMPTE）

或者垂直边沿或者水平边沿的滤波操作执行如下

$$\begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \end{bmatrix} = \left[\begin{bmatrix} 7 & 0 & 0 & 1 \\ -1 & 7 & 1 & 1 \\ 1 & 1 & 7 & -1 \\ 1 & 0 & 0 & 7 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} r_0 \\ r_1 \\ r_0 \\ r_1 \end{bmatrix} \right] \gg 3 \tag{8.11}$$

这里，x_i，y_i 和 r_i 分别表示原始像素、滤波像素和取整的值。对于奇数列/行，$r_0=4$ 且 $r_1=3$，假定块内编号从 1 开始。对于偶数列/行，$r_0=3$ 且 $r_1=4$。

式（8.11）的逆矩阵应该用在编码器端。注意，该矩阵应该实现一种低通滤波，将原始数据分布到两个相邻块的边沿上，如上面解释的那样。如果在量化过程

中损失了一个块的一些纹理，那么由于解码器的逆分布运算，丢失的纹理可以被获取回来。交换数据的时候，帧内编码块和帧间编码块不应该交换的（也就是说，帧间编码是针对残差数据的）。因此，当两个块的数据几乎都饱和在 255 时，由于运算的线性特性滤波可能产生溢出。为了避免溢出，标准对 OLT 定义了 128 的平移（加到块的每个像素上）然后箝位到［0，255］的范围。

8.4.5 每个宏块的 4 个运动矢量

在 P 和 B 帧的首部指明了运动矢量模式（MNMODE，VLC）。对于 P 帧，定义了 5 种模式：1MV、1MV-半像素 MV、1MV-半像素-双线性、混合-MV、和亮度补偿。对于简单档次，不使用亮度补偿模式。对于高级档次，亮度补偿意味着参考帧的像素在用它们作为当前帧预测器之前要重新映射。如果 MVMODE 给出混合-MV 模式，那么就会出现 MVTYPEMB 来指示用位平面编码的 1MV 或 4MV，如图 8.10 所示。位平面中的 0 值表示宏块以 1MV 模式编码，而 1 值表示宏块以 4MV 模式编码。SKIPMB 也指出在使用位平面编码方法时跳过哪个 MB。跳跃模式意味着该宏块没有运动矢量数据且没有残差数据。MVTAB 是一个 2bit 的语法元素，规定了四个表的哪一个用于解码宏块层的 MVDATA。

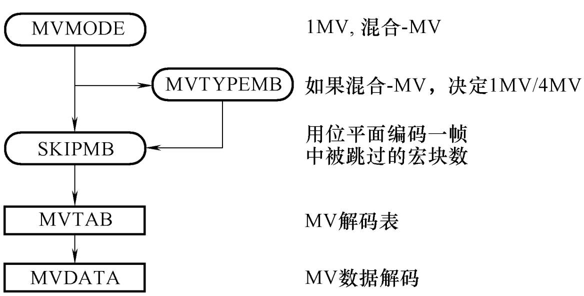


图 8.10 在简单/主档次 P 帧层中 MV 解码的语法流程图

MVDATA 包含宏块的运动矢量差分信息。在所有的 1MV 和 4MV 情况下，通过预测值加上差分值重建亮度分量的运动矢量，计算如下：

$$mv_x = (dmv_x + predictor_post_x)\ \mathrm{smod}\ range_x$$

$$mv_y = (dmv_y + predictor_post_y)\ \mathrm{smod}\ range_y$$

式中，range 表示由 MVRANGE 定义的水平或垂直方向上运动矢量的范围，predictor_post 表示水平或垂直的运动矢量预测值，dmv 表示差分运动矢量。Smod 意思是带符号模数操作符，允许重建矢量可以具有负值。$(A\ \mathrm{smod}\ b) = ((A+b)\&(2b-1))-b$，其中 b 是 2 的幂，通常是在感兴趣的范围，在 $-b$ 和 $b-1$ 之间。这里的 & 表示位与操作符。

宏块的运动矢量预测值（MVP，Motion Vector Predictor）从三个相邻宏块或块的候选运动矢量中选出，如图 8.11 所示。上面的两种方式用于 16×16 块运动

补偿模式。通常情况下，第一种方式用于选择候选运动矢量。如果当前宏块的位置在一行的最后，那么是得不到宏块 B 的。在此情况下，用第二种方式来选择候选运动矢量。下面两种方式用于混合大小的运动补偿模式，其中当前块用 16×16块的运动补偿，相邻块用 8×8 块的运动补偿。对于从相邻宏块中选择候选运动矢量它们是相似的，除了最近的 8×8 块是从每个宏块的四个 8×8 块中选出。

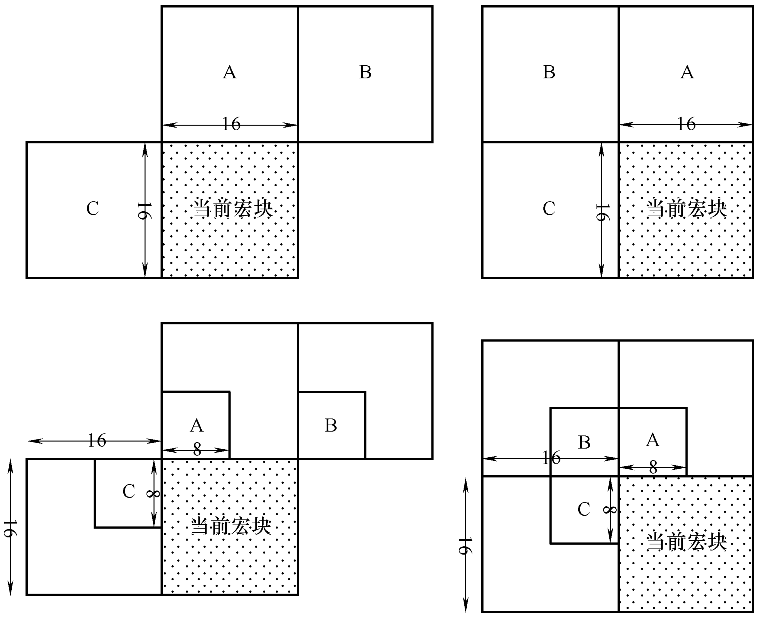


图 8.11　P 帧中 1MV（上面两个）和混合 MV（下面两个）的候选运动矢量预测器［C7］（2006 SMPTE）

8×8 块的 MVP 选自三个相邻宏块或块的候选运动矢量，如图 8.12 所示。该方式用一个宏块内每个块的候选运动矢量来描述——块 0（左上块）、块 1（右上）、块 2（左下）或块 3（右下）。对于某些特殊情况，由于在相邻块得不到 MV 数据就会选择某些不同的方式。对于宏块是一行中的第一宏块的情况，与该行中的剩余块相比，块 0 的预测器 B 要进行不同的处理。在此情况下，预测器 B 来自当前宏块顶上方宏块中的块 3 而不是当前宏块左上宏块中的块 3。如果宏块是列中的第一宏块，那么块 0 和 2 的预测器 C 设为 0。

8.4.6　Y 分量的 1/4 像素运动补偿

用于大多数视频编码标准的运动矢量在二维矢量空间中表示，为运动补偿过程提供从当前帧或场的坐标位置到参考帧或场坐标的偏移。使用运动矢量是为了通过降低预测误差提高样值预测的效率。在 VC-1 编码器中，预测过程利用运动矢量提

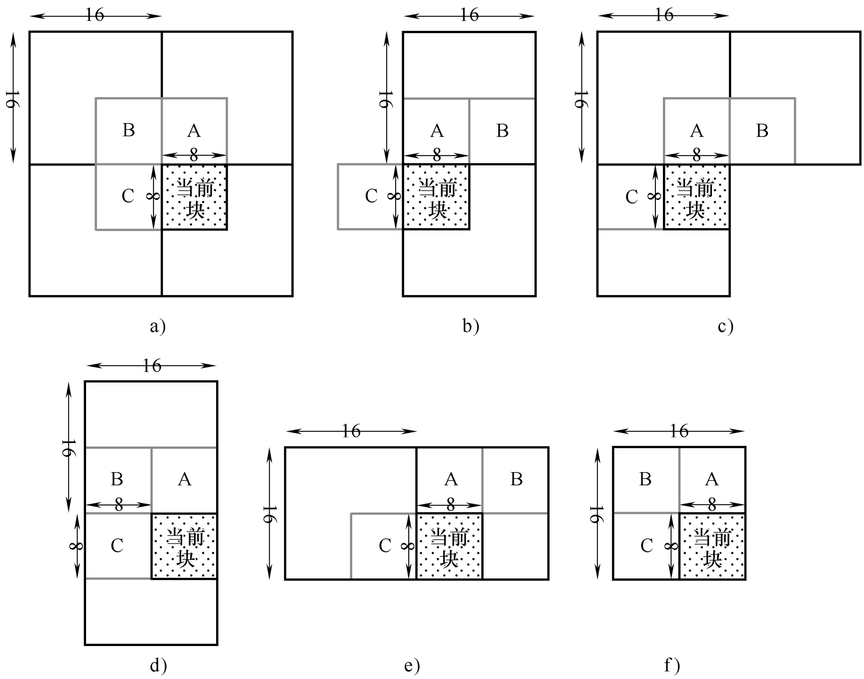


图 8.12　4MV 的 MV 预测器（2006 SMPTE）

a）如果不是行中首个宏块，块 0 的预测器　b）如果是行中首个 MB，块 0 的预测器
c）如果不是行中最后的宏块，块 1 的预测器　d）如果是行中最后的宏块，
块 1 的预测器　e）块 2 的预测器　f）块 3 的预测器

供对包含已解码样值的过去和/或未来参考帧或场的偏移。

一个视频编译码器的效率与生成一组好预测值的运动补偿器的能力密切相关。运动补偿的质量右四个因素决定：

1）亚像素分辨率；

2）预测区域的大小；

3）搜索窗口的大小；

4）用于插值的滤波器。

尽管亚像素分辨率改善了预测的质量，但走向更精细像素分辨率的好处被更高精度地编码运动矢量增加的代价所抵消。在低比特率时，因为用于编码运动矢量的比特百分比是相当大的，所以运动矢量的较高精度是理所应当的。在标准的视频编译码器存在的档次中，运动矢量分辨率或者是 1/2 或者是 1/4 像素。

影响生成好预测值能力的第二个因素是预测区域的大小。通常在较为陈旧的格式中，一个运动矢量用于一个宏块，它是亮度平面中的一个 16×16 的像素区域。MPEG-4 允许对 16×16 或 8×8 的块具有运动矢量；对于每个被编码的宏块都做这种选择。H.264/AVC 允许运动矢量的参考区域小到 4×4。尽管这种程度的自由可

以用于高比特率上，但较小的区域会给编译码器施加较高的计算开销。带有随机分布运动矢量的较小块引起对缓存的访问次数，而且它们需要更多的按像素进行的滤波步骤。这样，解码器的计算复杂度相应地提高了。VC-1 默认使用 16×16 的块，但允许帧内有 8×8 大小的块，该帧指明包含混合运动矢量的分辨率［C29］。

第三个因素，搜索窗口的大小（见图 8.13），决定了预测中的（$M\times N$）块的运动矢量（水平和垂直）范围。较大的搜索窗口可以找到最佳的匹配，但要涉及更多的计算。

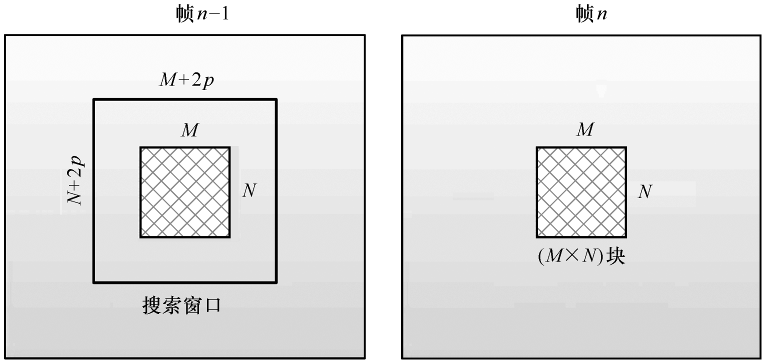


图 8.13 在帧 $n-1$ 中搜索运动矢量的搜索窗口

最后，用于生成亚像素预测值的滤波器是运动补偿质量的第四个关键决定因素。较短的滤波器计算较为简单但频率响应较差，而且反过来又受到噪声的影响。较长的滤波器用到更多的像素，计算起来更难实现。而且，具有局部和暂态特性的图像使用长滤波器趋向于发生抖动。VC-1 通过使用两组运动补偿滤波器平衡了这些考虑。第一组是四抽头的近似双三次滤波器，第二组是两抽头的双线性滤波器。图 8.14 中像素的双线性内插计算取决于像素位置。A，B，C 和 D 是整数像素。h 位于水平半像素和垂直半像素的位置上。g 位于水平半像素和垂直 1/4 像素的位置上。尽管双线性内插器是为 1/4 像素 MV 分辨率定义的，但只有半像素运动用于亮度块，1/4 像素分辨率只用于色差块。对于半像素情况，双线性内插由以下公式进行

$$
\begin{aligned}
b &= (A+B+1-RND) \gg 1 \\
f &= (A+D+1-RND) \gg 1 \qquad (8.12)\\
h &= (A+B+C+D+2-RND) \gg 2
\end{aligned}
$$

式中，RND 是帧级取整控制值。简单/主档次和高级档次之间的策略是不同的。对于简单/主档次，在每个 I 和 BI 帧，RND 的值重置为 1。注意，BI 帧是一帧左右宏块进行帧内编码的 B 帧。每个 P 帧的 RND 在 0 和 1 之间切换。B 帧的 RND 与以前的锚定帧（I 或 P）保持相同。对于高级档次，RND 从帧首部的 RNDCTRL 语法元素导出。

应用于所有情况的一般规则如下所示。数组 F 和 G 是滤波器的系数。给定

$F[\ ]=\{4, 3, 2, 1\}$ 和 $G[\ ]=\{0, 1, 2, 3, 4\}$，内插值 p 给出如下：

$$p=(F[x]F[y]A+F[x]G[y]B+G[x]G[y]C+G[x]F[y]D+8-RND) \gg 4 \tag{8.13}$$

式中，x 和 y 是水平（从左到右）和垂直（从下而上）方向上的亚像素平移。在图 8.14 所示的由四个像素框定的原点在 A 处的区域里，它们的值的范围是 0～4。例如，对于 g 的亚像素偏移是 $x=2$ 而 $y=1$。式（8.13）中一般的规则产生了半像素内插的在式（8.12）中的特殊情况。

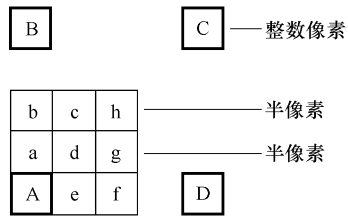


图 8.14　双线性插值的整数像素、半像素和 1/4 像素位置

双三次内插滤波器抽头定义如下：

$$T_{1/2}[\ ]=\{-1,9,9,-1\} \text{ for 1/2 pel shift},$$
$$T_{1/4}[\ ]=\{-4,53,18,-3\} \text{ for 1/4 pel shift},$$
$$T_{3/4}[\ ]=\{-3,18,53,-4\} \text{ for 3/4 pel shift}. \tag{8.14}$$

图 8.15 中示例的双三次内插计算如下：

$$\begin{aligned}
dd&=(-4A+53D+18H-3K+32-r)\gg 6,\\
ii&=(-A+9D+9H-K+8-r)\gg 4,\\
nn&=(-3A+18D+53H-4K+32-r)\gg 6,\\
a&=(-4C+53D+18E-3F+32-r)\gg 6,\\
b&=(-C+9D+9E-F+8-r)\gg 4,\\
c&=(-3C+18D+53E-4F+32-r)\gg 6,\\
M&=(-4S+53C+18G-3U+32-r)\gg 6,
\end{aligned} \tag{8.15}$$

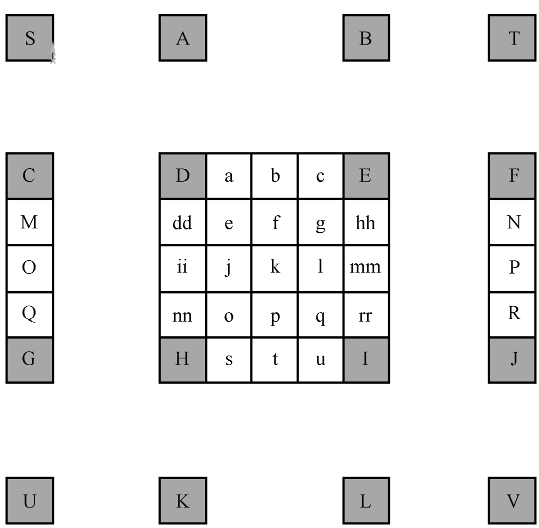


图 8.15　双三次内插的整数像素、半像素和 1/4 像素位置

式中，对于垂直方向，$r = 1 - \mathrm{RND}$；而对于水平方向，$r = \mathrm{RND}$。垂直滤波输出在取整之后输入到水平滤波环节。

8.5　主档次

8.5.1　CbCr 的 1/4 像素运动补偿

色度分量总是利用双线性内插法，而且 VC-1 目前只定义了一种色度格式 YCbCr 4：2：0。因为色度运动矢量隐含地来自于相同位置上的亮度运动矢量，精度有限且有简化的余地。另外，色度通道具有很强的低通分量。VC-1 用双线性滤波器实现色度的运动内插。一般而言，通过将相同位置的亮度运动矢量除以 2 再取整到 1/4 像素的位置得到色度运动矢量。另外，序列级上有个 1bit 的域控制着色度运动矢量的取整。如果设置了该比特，那么偏移了四分之一像素的色度运动矢量就被取整到最近的全像素位置——事实上，对于色度运动矢量只允许 1/2 和全像素的位置。这种模式的目的是为了解码器的速度优化。该优化的动机是在不同精度偏移上进行内插的复杂度的显著差异：①整数像素；②半像素；③（x 和 y）至少有一个坐标在 1/4 像素位置上；④两个坐标都在 1/4 像素位置上。以上 a∶b∶c∶d 大约是 1∶4∶4.7∶6.6。通过使用这种模式，就可以做到支持①和②，这样就削减了解码时间。因为这只是对色度内插进行的操作，编码和质量损失（尤其是主观质量）都可以忽略［C2］。

有一个面向解码软件设计的叫作 FASTUVMC 的 Fast UV 运动补偿标志，旨在降低内插运算的复杂度，因为内插环节是解码软件中计算最复杂的部分。VC-1 编码器可以利用 FASTUVMC 模式，这种模式不会生成 1/4 像素精度的色度运动矢量。如果 FASTUVMC == 1，偏移 1/4 像素的色差运动矢量就被取整到最近的半像素或全像素的位置。如果 FASTUVMC == 0，那么对色差分量就没有任何取整和滤波操作。对于简单档次而言，FASTUVMC 设置为 1。

8.5.2　起始码

起始码是一个嵌入在比特流中的 32bit 的码，它是唯一的且指示了比特流数据单元（BDU，Bitstream Data Unit）的开头。BDU 是一个压缩数据的单元，在相同的层次级别上可以在语法方面（语法解码后）独立于其他信息。BDU 可以是，例如，一个序列首部、一个入口点首部，一个编码帧或分片。起始码包含一个独一无二的三字节起始码前缀（SCP，Start Code Prefix）0x000001，和一个一字节的起始码后缀（SCS，Start Code Suffix），它用于识别起始码后面 BDU 的类型。例如，一帧之前的起始码后缀是 0x0D，不同于一个分片之前的起始码后缀 0x0B。块和宏块不是 BDU，因而没有起始码也不必字节对齐［C4］。

起始码模拟预防（原文：emulation prevention——译者注）用于表 8.7 中 VC-1_SPMP_PESpacket_PayloadFormatHeader（ ）结构里的所有字节，以及该结构之后的 PES 包有效荷载字节中，以避免起始码出现在 PES 包有效荷载的任何其他位置上。VC-1 定义了一种利用字节填充的封装机制。用这种机制封装的 BDU 被称为封装 BDU（EBDU，Encapsulated BDU），而被编码器封装之前或者被解码器提取之后的压缩数据被称为是原始 BDU（Raw BDU）。

表 8.7　VC-1_SPMP_PESpacket_PayloadFormatHeader（ ）的结构［C12］（2007 SMPTE）

语　法	比特数	格　式
VC－1_SPMP_PESpacket_PayloadFormatHeader（ ）		
{		
start_code	32	0x0000010F or 0x0000010D
if（start_code ==0x0000010F）		
{	16	
frame_width	16	Uimsbf
frame_height	32	Uimsbf
STRUCT_SEQUENCE_HEADER_C（ ）	32	Bslbf
start_code		0x0000010D
}		
else if（start_code ==0x0000010D）		
{		
}		
}		

8.5.3　扩展的运动矢量

VC-1 中默认运动矢量的范围是［－64，63.f］×［－32，31.f］，其中 f 对于 1/4 像素的运动分辨率是分数 3/4，对于 1/2 像素的运动分辨率是 1/2。换言之，对于默认的 1/4 像素运动模式是沿着水平（X）轴的［－64，$63\frac{3}{4}$］，以及沿着垂直（Y）轴的［－32，$31\frac{3}{4}$］。对于简单档次，应该使用默认范围。对于主档次和高级档次，扩展运动矢量模式由标志 EXTENDED_MV 指示出来。如果 EXTENDED_MV ==0，使用默认范围。如果 EXTENDED_MV ==1，用标志 MVRANGE 给出扩展运动矢量范围，如表 8.8 所定义。

如果 EXTENDED_MV ==1，扩展的差分运动矢量范围标志（EXTENDED_DMV）就会出现在入口点片段内 P 帧和 B 帧的帧层里。

表 8.8　VC-1 中 MVRANGE 指示的运动矢量范围

MVRANGE VLC	运动矢量范围（水平×垂直）
0b（默认）	[−64，63. f] ×[−32，31. f]
10b	[−128，127. f] ×[−64，63. f]，
110b	[−512，511. f] ×[−128，127. f]，
111b	[−1024，1023. f] ×[−256，255. f]，

© 2006SMPTE

表 8.9　如果 MAXBFRAMES >0 主档次的帧类型 [C7]

PTYPE VLC	帧　类　型
1b	P
01b	I
00b	B 或 BI

（2006　SMPTE）

8.5.4　环路滤波器

重叠平滑之后，实施环路滤波来进一步基于帧/场降低块效应。分片数据也被滤波，每个分片的环路滤波器独立执行。如果语法元素 LOOPFILTER ==1，那么要进行环路滤波。通过对可能具有块效应的块边沿每一侧的 1 个像素进行滤波来平衡滤波的程度和复杂度。环路滤波器包含在 VC-1 编码/解码环路之内，因而叫作环路滤波器（ILF，in-loop Filtering）。注意，在环路之外可能出现较重的滤波。

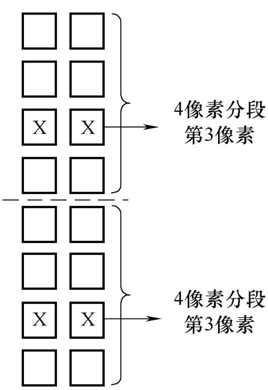


图 8.16　用于 ILF 的两个 4 像素分段 [C7]
2006　SMPTE

对于 I，B 和 BI 帧，ILF 在所有的 8×8 的块边界上执行。因为一行或一列中被滤波的连续像素的最小个数是 4，而且一行或一列中像素总数总是 4 的倍数，所以在 4 像素段上执行的 ILF 如图 8.16 给出。在 4 个像素中的第 3 个像素首先被滤波，在第二条路径中剩余的 3 个像素由不同的算法进行滤波（见图 8.17）。两个边界像素通过图 8.18 所示的滤波操作加以修改。布尔值‘filter_other_3_pixels’定义了片段内剩余的 3 个像素对是否也要滤波。如果 filter_other_3_pixels == TRUE，那么其他 3 个像素对要进行滤波。否则，滤波操作前往下一个 4 像素段。

对于 P 帧，使用同样的滤波会有一些例外。块可以是帧内编码或帧间编码的。帧内编码块应该用 8×8 的逆变换来重建样值，而帧间编码块用 8×8，8×4，4×8，或 4×4 的逆变换。变换块或子块之间的边界像素是要滤波的，除非发生以下例外情况：当边界任一侧的变换块（或子块）都是帧间编码的，且当两个块（或子块）所有的变换系数都等于零时，不会执行滤波 [C7]。在此情况下，不滤波的原因是

```
//第3像素滤波

filter_other_3_pixels = TRUE
a0 = (2*(p3 - p6) - 5*(p4 - p5) + 4) >> 3;
if (abs(a0) < PQUANT)
  a1 = (2*(p1 - p4) - 5*(p2 - p3) + 4) >> 3;
  a2 = (2*(p5 - p8) - 5*(p6 - p7) + 4) >> 3;
  a3 = min(abs(a1), abs(a2));
  if (a3 < abs(a0))
    d = 5*((sign(a0) * a3) - a0)/8;
    clip = (p4 - p5)/2;
    if (clip == 0)
      filter_other_3_pixels = FALSE
    else
      if (clip > 0)
        if (d < 0)
          d = 0;
        if (d > clip)
          d = clip;
      else
        if (d > 0)
          d = 0;
        if (d < clip)
          d = clip;

      p(4) = p4 - d;
      p(5) = p5 + d;
  else
    filter_other_3_pixels = FALSE;
else
  filter_other_3_pixels = FALSE;
```

```
//第1,第2和第4像素滤波

a0 = (2*(p3 - p6) - 5*(p4 - p5) + 4) >> 3;

if (abs(a0) < PQUANT)
  a1 = (2*(p1 - p4) - 5*(p2 - p3) + 4) >>
  3;
  a2 = (2*(p5 - p8) - 5*(p6 - p7) + 4) >>
  3;
  a3 = min(abs(a1), abs(a2));
  if (a3 < abs(a0))
    d = 5*((sign(a0) * a3) - a0)/8;
    clip = (p4 - p5)/2;
    if (clip > 0)
      if (d < 0)
        d = 0;
      if (d > clip)
        d = clip;

      p(4) = p4 - d;
      p(5) = p5 + d;
    else if (clip < 0)
      if (d > 0)
        d = 0;
      if (d < clip)
        d = clip;

      p(4) = p4 - d;
      p(5) = p5 + d;
```

图 8.17　对于第 3 像素（左）和第 1，第 2 和第 4 像素（右）的滤波算法［C7］（2006 SMPTE）

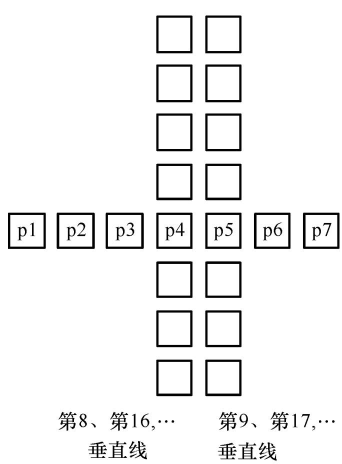


图 8.18　两条垂直边界线环路滤波的例子［C2］（2004 Elsevier）

由于复制过来一个已经滤波的参考帧。

8.5.5 动态分辨率的改变

动态分辨率改变是指主档次中通过语法元素 RESPIC（图像分辨率，2bit）用因数 2 缩放编码帧大小，以及在高级档次入口点首部中通过传输的编码大小的语法元素用一个任意的缩放因数缩放编码帧大小。MULTIRES（多分辨率）标志指示帧是否可在比规定帧分辨率更低的分辨率上进行编码。分辨率改变应该只允许在 I 帧上进行。如果 MULTIRES ==1，那么帧级语法元素 RESPIC 就会出现，它给出了这一帧的分辨率。RESPIC 可以出现在 I 和 P 帧中。如果它出现在 P 帧首部，那么 RESPIC 语法元素与前面最近的 I 帧的 RESPIC 相同。换言之，所有的 P 帧以与前面最近的 I 帧相同的分辨率进行编码。在 B 帧的情况中，分辨率等于两个参考帧的分辨率。如果 RESPIC 指示了下采样分辨率，那么新一帧的维数计算如图 8.19 所示。如果解码帧是下采样分辨率之一，那么它在显式之前会被上采样到完整的分辨率。

```
X = 16 * ((CodedWidth + 15) / 16)
Y = 16 * ((CodedHeight + 15) / 16)
x = new horizontal resolution
y = new vertical resolution

hscale = horizontal scaling factor (0 = full resolution, 1= half resolution)
vscale = vertical scaling factor (0 = full resolution, 1= half resolution)

x = X
y = Y
if (hscale == 1)
{
        x = X / 2
        if ((x & 15) != 0)
                x = x + 16 – (x & 15)
}
if (vscale == 1)
{
        y = Y / 2
        if ((y & 15) != 0)
                y = y + 16 – (y & 15)
}
```

图 8.19 下采样多分辨率中帧维数的计算［C7］（2006 SMPTE）

8.5.6 B 帧

B 帧的出现由序列层中的语法元素 MAXFRAMES 指出。如果 MAXBFRAMES >0，

那么就会出现若干B帧。帧类型可以是P、I、B或BI，如表8.10所定义。如果PTYPE等于B帧，那么就会定义B帧分数（BFRACTION）值。BFRACTION给出一个分数，它可能在一个0和1之间的分数值有限集合中取值，表示其锚定帧形成的时间间隔内B帧的相对时间位置。这样分数值就是一个表8.10中的缩放因子，它将用于缩放相同位置上的运动矢量来获得“直接”运动矢量。

表8.10 在0和1之间的BFRACTION值［C7］

BFRACTION VLC	分　　数	BFRACTION VLC	分　　数
000b	1/2	1110101b	2/7
001b	1/3	1110110b	3/7
010b	2/3	1110111b	4/7
011b	1/4	1111000b	5/7
100b	3/4	1111001b	6/7
101b	1/5	1111010b	1/8
110b	2/5	1111011b	3/8
1110000b	3/5	1111100b	5/8
1110001b	4/5	1111101b	7/8
1110010b	1/6	1111110b	Reserve
1110011b	5/6	1111111b	BIpictur
1110100b	1/7		

（2006 SMPTE）

如果一个锚定帧（I或P帧）被编码为一个跳过帧，那么它被当作一个P帧来处理。在此情况下，两个锚定帧对于介入其中的B帧都是相同的。例如，如果帧按如下显示顺序进行编码：

I0 B1 P2 B3 P4 B5 S6（I0 P2 B1 P4 B3 S6 B5按编码顺序）其中S6是跳过帧，那么这实际上处理为：

I0 B1 P2 B3 P4 B5 P4。

跳过帧的运动矢量设置为零，且运动矢量的计算基于来自后续锚定帧和预定义缩放逻辑，叫作直接模式运动矢量。如果相同位置的宏块是帧内编码的，那么直接模式运动矢量设置为（0，0）。

迄今为止，已经讨论了只在帧层里面主档次的B帧出现的显著更新。进一步的更新在随后的宏块层中。

8.5.7 自适应宏块量化

对于主档次和高级档次引入了增强的宏块量化方法。在主档次中，仅仅使用渐进模式，而在高级档次中则定义了隔行模式。渐进模式表明逐帧进行编码，而隔行模式表明或者按帧或者按场进行编码。每一帧可编码为I帧（帧内编码）编码，BI帧（只包括帧内宏块的B帧），P帧（前向预测编码），跳过帧，或B帧（前向和

后向预测编码）。I 帧中的所有宏块都是帧内编码的。P 帧的宏块可以进行帧内编码或者帧间编码，取决于局部特性。同样的原则适用于 B 帧。

宏块层的首部相应于它的编码模式，见表 8.11。渐进帧编码模式中的 I 帧包括编码块模式（CBPCY）和宏块层中允许或不允许（ACPRED）的交流预测。对于高级档次，OVERFLAGMB 用于指明是否要对相邻块之间的边缘像素执行重叠滤波。MQDIFF 给出宏块量化器的差分值；ABSMQ 是宏块量化器的绝对值，意思是 MQUAT 等于 ABSMQ。

表 8.11　宏块层的首部

帧模式	帧/场	帧类型	宏块首部	附注
渐进	帧	I（BI）	CBPCY，ACPRED	
			CBPCY，ACPRED，OVERFLAGMB，MQDIFF，ABSMQ	高级档次
		P	MVMODEBIT，SKIPMBBIT，MVDATA，HYBRIDPRED，ACPRED，CBPCY，MQDIFF，ABSMQ，TTMB	1MV 模式
			MVMODEBIT，SKIPMBBIT，CBPCY，BLKMVDATA，MVDATA，HYBRIDPRED，MQDIFF，ABSMQ，ACPRED，TTMB	4MV 模式
		B	MVMODEBIT，SKIPMBBIT，CBPCY，BLKMVDATA，MVDATA，HYBRIDPRED，MQDIFF，ABSMQ，ACPRED，TTMB	
隔行	帧	I（BI）	FIELDTX，CBPCY，ACPRED，OVERFLAGMB，MQDIFF，ABSMQ	
		P	SKIPMB，MBMODE，FIELDTX，CBPPRESENT，CBPCY，ACPRED，MQDIFF，ABSMQ	帧内宏块
			SKIPMB，MBMODE，CBPCY，2MVBP/4MVBP，MVDATA，MQDIFF，ABSMQ，TTMB	帧间宏块
		B	SKIPMB，MBMODE，FILEDTX，CBPPRESENT，CBPCY，ACPRED，MQDIFF，ABSMQ	帧内宏块
			SKIPMB，MBMODE，DIRECTBBIT，BMVTYPE，MVSW，CBPCY，2MVBP/4MVBP，MVDATA，MQDIFF，ABSMQ，TTMB	帧间宏块
	场	I（BI）	CBPCY，ACPRED，OVERFLAGMB，MQDIFF，ABSMQ	
		P	MBMODE，MQDIFF，ABSMQ，ACPRED，CBPCY	帧间宏块
			MBMODE，MVDATA，HYBRIDPRED，CBPCY，MQDIFF，ABSMQ，TTMB	1MV 宏块
			MBMODE，MVDATA，HYBRIDPRED，CBPCY，MQDIFF，ABSMQ，TTMB	4MV 宏块
		B	MBMODE，MQDIFF，ABSMQ，ACPRED，CBPCY	帧内宏块
			MBMODE，FORWARDBIT，BMVTYPE，INTERPMVP，BMV1，BMV2，CBPCY，MQDIFF，ABSMQ，TTMB	1MV 宏块
			MBMODE，FORWARDBIT，BMVTYPE，INTERPMVP，BMV1，BMV2，CBPCY，MQDIFF，ABSMQ，TTMB	4MV 宏块

* 高级档次支持隔行模式

ACPRED 是个 1 比特的语法元素，出现在所有的 I 帧和 BI 帧宏块以及 P 帧和 B 帧的帧内编码的宏块中。如果 ACPRED == 1，那么交流系数预测用于 8 × 8 的块。解码块顶行或最左列的交流系数被视为预测块相应的行或列系数的差分值，如图 8.20 所示。对每个块，直流预测器所选的方向会被用于交流预测器。紧邻的上方/左侧块中交流系数的首行/列用作当前块交流系数首行/列的预测器。如果在预测方向上不存在块，例如图像边界，那么 7 个系数的预测值都设置为零。VC-1 采纳了一种简单的一阶预测器。

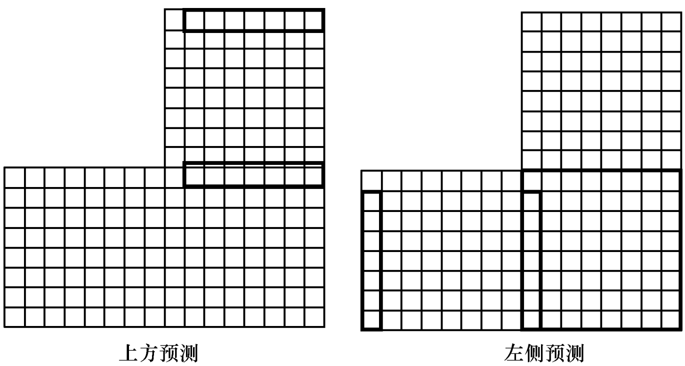


图 8.20　上方或左侧的交流预测系数［C7］（2006 SMPTE）

在 P 帧，MVMODEBIT 用于定义若干运动矢量（如果 MVMODEBIT == 0，那么 1 个运动矢量。如果 MVMODEBIT == 1，那么 4 个运动矢量。）SKIPMRBIT 指示宏块是否被跳过。MVDATA 只在 SKIPMBBIT == 0（不跳过）是出现。HYBRIDPRED 是个每个运动矢量 1 比特/的语法元素，指示要用哪一个运动矢量预测器。最终的运动矢量通过组合顶部（预测器 A）和/或左侧（预测器 C）的运动矢量得到。跳过的宏块会有 HYBRIDPRED，标识着去解码宏块的运动矢量。

8.5.8　亮度补偿

对于主档次，亮度补偿由 P 帧首部的语法元素 MVMODE 给出。如果 MVMODE == 0001b，就执行亮度补偿。如果指示了亮度补偿，那么 MVMODE2，LUMSCALE 和 LUMSHIFT 语法元素随后出现在帧层。P 帧中出现 MVMODE2 是为了实现不同类型 MV 模式的亮度补偿。为 MVMODE2 定义的 VLC 码表取决于 PQUANT 的值。四种不同的模式是 1-MV、混合 MV、1MV-半像素、和 1MV-半像素-双线性模式。如果 MVMODE 指示亮度补偿，为了建立用于重新映射参考帧像素的查找表，还定义了 LUMSCALE 和 LUMSHIFT，如图 8.21 所解释的那样。参考帧的 Y 分量用 LUTY［ ］表重新映射，而 Cb 和 Cr 分量用 LUTUV［ ］表来重新映射。

```
if (LUMSCALE == 0)
{
      iScale = -64;
      iShift = 255 * 64 - LUMSHIFT *2 * 64;
      if (LUMSHIFT > 31)
            iShift += 128 * 64;
}
else {
      iScale = LUMSCALE + 32;
      if (LUMSHIFT > 31)
            iShift = LUMSHIFT * 64 - 64 * 64;
      else
            iShift = LUMSHIFT * 64;
}
// build LUTs
      for (i = 0; i< 256; i++)
      {
            j = (iScale * i + iShift + 32) >> 6;
            if (j > 255)
                  j = 255;
            else if (j < 0)
                  j = 0;
                  LUTY[i] = j;
                  j = (iScale * (i - 128) + 128 * 64 + 32) >>6;
            if (j > 255)
                  j = 255;
            else if (j < 0)
                  j = 0;
            LUTUV[i] = j;
}
```

图 8.21　亮度补偿的查找表计算［C7］（2006 SMPTE）

8.5.9　范围调节

如果序列级的 RANGERED 标识设置为 1，而且通过高级档次中的语法元素 RANGE_MAPY 和 RANGE_MAPUV 进行范围映射，那么范围调节是指通过主档次的 RANGEREDFRM 语法元素用因数 2 来减小范围。如果 RANGERREDFRM = = 1，就对帧减小范围。对于一个 B 帧或 BI 帧，RANGERREDFRM 的值等于显示顺序中随后锚定帧的相应值。

如果 RANGEREDFRM = = 1，编码器的像素要缩小，解码器的像素要放大，保证当前重建帧具有原始范围。当前 I 和 P 帧解码的放大过程按照下列公式执行：

$$\begin{aligned} Y[n] &= \text{clip}((Y[n]-128)*2+128); \\ C_b[n] &= \text{clip}((C_b[n]-128)*2+128); \\ C_r[n] &= \text{clip}((C_r[n]-128)*2+128); \end{aligned} \tag{8.16}$$

8.6 高级档次

8.6.1 比特流结构

VC-1解码器用发送自编码器的元数据初始化它的解码过程。高级档次的序列层数据结构如表8.12所定义，它开始于码值0xC5，后接若干编码帧和相关参数。高级档次的STRUCT_C包括且只保留PROFILE（4bit）信息。其他参数都在后来的层中定义：入口点层和帧层。STRUCT_A包括8个连续的零字节。STRUCT_C包括LEVEL、CBR、一些保留比特和FRAMERATE。

表8.12 高级档次的序列层数据结构［C7］

31～24（1B）	23～0（3B）
0xC5（8B）	NUMFRAMES（24bit）
0x00000004（4B）	
STRUCT_C包括PROFILE（4bit）and Reserved7（28bit）	
STRUCT_A包括8个连续的零字节	
0x0000000C（4B）	
STRUCT_B包括LEVEL（3B）、CBR（1bit），RES1（4bit），RES2（56bit），FRAMERATE（4B）	

（2006 SMPTE）

8.6.2 隔行扫描

一帧包含一个视频信号空间信息的所有行。对于渐进式视频，这些包含样值的行从一个时刻开始持续经过连续的多行一直到达帧底。对于隔行视频，一个场是一帧各个交替出现的行的组合。因而，一帧包括两个场，顶场和底场。在隔行场编码模式中，隔行帧的两个场单独编码。在隔行帧编码模式中，隔行帧的两个场一起编码。

如果序列级别1bit的语法元素INTERLACE设定为1，那么后续帧进行隔行编码。在帧层内，给出了大小可变的元素——帧编码模式（FCM，Frame Coding Mode），来表明帧编码为渐进式、隔行场式还是隔行帧式。UV采样格式（UVSAMP）是个1bit的语法元素，当序列级别的域INYERLACE ==1时它只出现在所有的高级档次的帧首部中。如果UVSAMP ==1，那么使用色差分量的渐进式亚采样。如果UVSAMP ==0，那么使用色差分量的隔行式亚采样。帧类型（PTYPE）或场类型（FTPYE）给出当前帧的重建：I，P，B，BI，I/I，I/P，P/I，P/P，B/B，B/BI，BI/B或BI/BI。

在隔行场图像中，帧首部中的顶场优先（TFF，Top-Field First）语法元素指出

两个场的时间顺序。如果 TFF ==1，第一个场是顶场。如果 TFF ==0，第二个场是顶场。在编码一帧的第一个场时，参考帧的顶和/或底场可能用作参考。超出参考帧边界的像素通过边缘像素的复制填充来生成。这是因为运动补偿的运动矢量可能会给出参考帧边界之外的像素位置。图 8.22 给出了水平/垂直、渐进/隔行、和帧/场像素填充。因而，顶场应该被填充。当第二个场使用参考帧的第二个场作为参考时，如同隔行帧的情况那样执行像素复制。其他位置诸如底行或右列以类似方式进行填充。

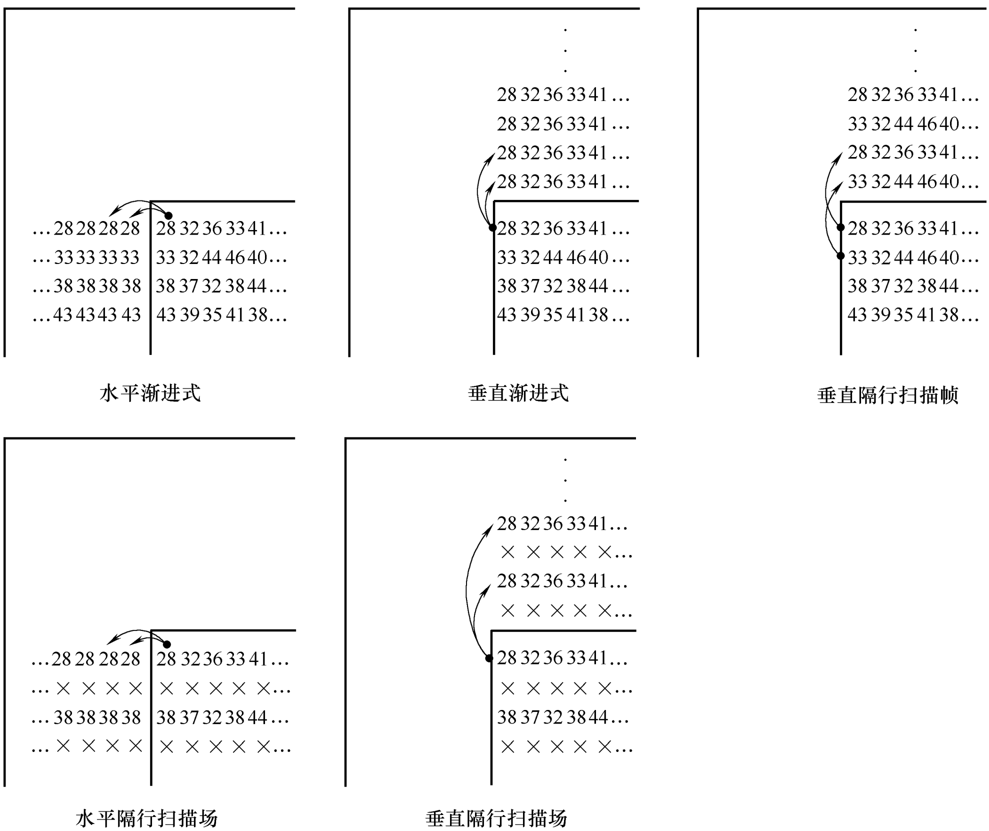
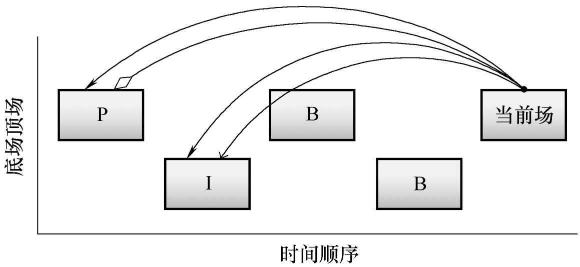


图 8.22　超界的复制填充示例 [C7]（2006 SMPTE）

一个隔行 P 场图像或者参考一个或两个以前解码的场。帧层中语法元素参考数目（NUMREF）是一个 1bit 的语法元素，指出当前场参考一个还是两个以前的参考场图像。如果 NUMREF ==1，那么当前隔行 P 场图像参考时间上最近（按显示顺序）的 I 或 P 场图像，如图 8.23 所示。如果 NUMREF ==0，那么当前隔行 P 场图像参考一个场。在此情况下，语法元素参考场指示器（REFFIELD）跟随帧层比特流中的语法元素 NUMREF。语法元素 REFFIELD 是个 1bit 的语法元素，它指出哪一个解码的场用作参考。如果 REFFIED ==0，那么时间上最近（显示顺序）的

I 或 P 场用作参考。如果 REFFIELD == 1，那么第二最近的 I 或 P 场图像用作参考。



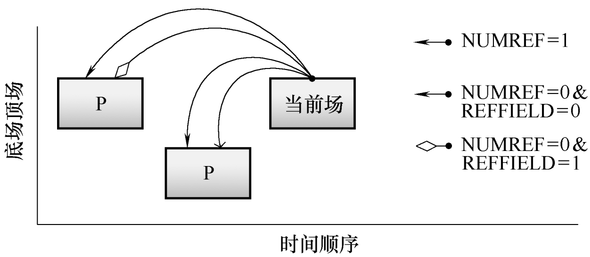


图 8.23　有 B 帧（上）和没有 B 帧（下）的两个参考隔行场图像示例［C2］（2004 Elsevier）

使用 B 帧时，前向帧和后向帧都用作运动补偿的参考。使用 B 场时，第一个 B 场可被用作第二个正被解码 B 场的参考。例如，如果第一个解码的场是顶场，那么那幅图像的底场就使用顶场作为运动补偿的参考。这是用相反极性（原文：opposite polarity——译者注）预测的情况。通过作为同极性场的前锚定帧的第二场以及从下一个锚定帧的第一和第二场可做出进一步的预测。NUMREF 和随后的 REFFIELD 不出现在 B 场图像中，因为 B 场总是用总共 4（前向的顶场和底场，后向的顶场和底场）个参考场来预测当前宏块。

8.6.3　序列级的用户数据

只有高级档次支持用压缩比特流携带带内用户数据（元数据）。在简单档次和主档次中，与序列相关的元数据可被运输层或其他途径传递到解码器。请参考 8.4.1 节。在高级档次中，序列相关的元数据是视频数据比特流的一部分，这部分简要描述了其语法和语义。

用户数据是一种允许携带数据的设施，该数据能提供必要的额外的用户特征。它能被用于字幕支持，例如［C4］。用户数据作为比特流数据单元（BDU，Bitstream Data Units）进行传输，可被包括在序列比特流、入口点、帧、场、或分片层。用户数据的起始码（每一层的值 0x0000011B ~ 0x0000011F）后接用户数据标识符，它是一个定长（32bit）的语法元素，显示出 SMPTE 注册的用户数据类型。然后传送若干字节的用户数据，并加上 flushing_byte（0x80）。

序列级别的用户数据位于序列首部之后起始码之前的比特流中，起始码表明下一个比特流数据单元的开端。图 8.24 给出了序列起始码（SEQ_SC）和序列首部（SEQ_HDR）之后的用户数据起始码（UD_SC）和用户数据（UD_DAT）两种可能的位置。上面的比特流描述了下一个 BDU 是一个入口点起始码（ENTRY_SC）后接一个入口点首部（ENTRY_HDR）的情况，而下面的比特流描述了下一个 BDU 是一个帧起始码（FRM_SC）后接一个帧数据（FRM_DAT）的情况。

SEQ_SC	SEQ_HDR	UD_SC	UD_DAT	ENTRY_SC	ENTRY_HDR	FRM_SC	FRM_DAT

a)

SEQ_SC	SEQ_SDR	UD_SC	UD_DAT	FRM_SC	FRM_DAT

b)

图 8.24　序列级用户数据的两种可能的位置［C7］（2006 SMPTE）

8.6.4　入口点层

在高级档次中，一个序列包含一个或多个入口点片段，每个入口点片段包含一系列帧，每个入口点片段的第一帧提供随机访问。一帧可分为多个宏块。一个分片包含一个或多个连续的宏块行。入口点和分片层只出现在高级档次中。每个入口点层片段的开头指明了一个随机访问点。在简单档次和主档次中，每个 I 帧是一个随机访问点。

8.6.5　显示元数据

VC-1 流包含的元数据不用于解码过程，而是传递到提供显示的过程用于采样的视频格式、样本纵横比、颜色空间等等。例如，解码过程的帧率可能低于目标显示帧率。在高级档次序列首部中，包含了帧率的域。首先，FRAMERATE_FLAG 和 DISPLAY_EXT 设置为 1 以定义显示帧率。然后通过 FRAMERATEEXP（16 比特）用帧率指示参数（FRAMERATEIND）来显式地（如果 FRAMERATEIND == 1）定义帧率，或者通过 FRAMERATENR（分子）和 FRAMERATEDR（分母）隐含地（如果 FRAMERATEIND == 0）定义。

当给出下拉（PULLDOWN == 1）信号时，解码帧可由一个 2bit 的语法元素 RPTFRM 进行重复显式。例如，一个 24 帧/s 的压缩比特流服务于 60 帧/s 的渐进显示，RPTFRM 对于连续帧在 1 和 2 之间交替取值，然后分别显示以 2 或 3 个帧周期显示帧。请注意，24～60 的帧率转换在 MPEG-2 标准中叫作 3∶2 下拉。当一个序列服务于隔行显示时，一对场的第一场可能要重复——这样就把场对（帧）的持续时间扩展到了三个场显示周期。

在所有的档次所有的渐进帧类型中，INTERPFRM 是一个 1bit 的语法元素。这

个比特不用在解码过程。它的目的是给显示过程提供一个提示，提示当前时间域适合进行时间插值。如果 INTERPFRM ==1，那么时间域适合帧插值。例如，显示过程可用内插法来提高显示帧率。如果 INTERPFRM ==0，那么当前时域（当前帧及其周围的帧）就不适合帧内插。对于差错恢复（取代丢失的帧）或者降低带宽（编码更少的帧以及内插到较高帧率来显示），这个比特是有用的。

在高级档次中，编码帧的宽度和高度可能在入口点首部或序列首部改变；而在简单档次/主档次中，帧大小从数据元素 HORIZ_SIZE 和 VERT_SIZE 中导出。序列首部域 MAX_CODED_WIDTH 和 MAX_CODED_HEIGHT 指出一个序列中最大的编码帧大小，入口点首部域 CODED_WIDTH 和 CODED_HEIGHT 指出入口点层中改变后的帧大小。当目标显示大小不同于编码帧大小时，序列首部就会使用域 DISP_HORIZ_SIZE 和 DISP_VERT_SIZE。样本纵横比可在高级档次中由 ASPECT_RATIO 域来定义。例如，如果显示大小是 720 像素宽 480 像素高，且使用了 10∶11（宽∶高）的样本纵横比，那么表示 4∶3 几何形状的帧的实际区域就会是 704 像素宽 480 像素高（480 ×4/3 ×11/10 =704）。摇摄（原文：pan——译者注）扫描区域是显示区域的一个子区域，例如在 16∶9 显示器上的 4∶3 子区域。如果入口点首部语法元素 PANSCAN_FLAG ==1，那么在高级档次帧首部中要给出指示。在此情况下，帧首部具有 PS_PRESENT。如果它被设置为 1，那么就用四个语法元素 PS_HOFFSET，PS_VOFFSET，PS_WIDTH 和 PS_HEIGHT 来定义帧内窗口的大小和位置。

当序列级标识 POSTPROCFLAG ==1 指示四种后处理模式（无后处理，去块效应，去振铃，和去块效应且去振铃）时，一个 2bit 的语法元素 POSTPROC 出现在高级档次的所有帧内。序列首部的后处理量化帧率（FRMRTQ_POSTPROC）和后处理量化比特率（BITRTQ_POSTPROC）为解码器提供信息，以便于解码器可以估计这个序列去块效应和去振铃操作的运算复杂度。这一估计与后处理器计算能力信息的结合，可被解码器用于决定是否启用或关闭 POSTPROC 指出的去块效应和去振铃操作

8.7 H.264 到 VC-1 的转码

视频转码是一个把视频从一种格式到另一种格式转换的操作。这一格式转换包含一系列的操作，例如降低比特率、从一种压缩格式到另一种的转换、改变视频容器格式或改变首部描述及其他。除了这个基本的格式转换之外，转码器还可被用做其他功能，诸如压缩视频编码参数的调整、空间和时间分辨率转换、和插入新信息诸如数字水印或公司徽标甚至是增强的差错恢复信息［T3］。

最简单的转码结构是将解码器和编码器直接级联起来，如图 8.25 所示。在此结构中，输入的源视频流 V_S 彻底解码然后再编码为具有想要的比特率或格式的目标视频流 V_T，转码过程不引入任何明显的视觉质量下降。这类实现涉及输入压缩

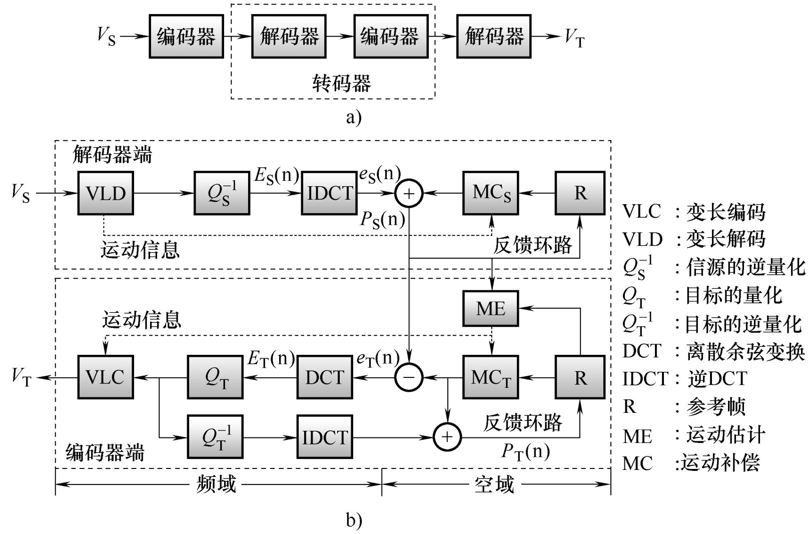


图 8.25　编码器和解码器级联模型［T3］（2005 IEEE）

a）块级框图　b）详细框图

视频流的彻底解码和重新编码。它必须在重新编码之前进行彻底的解码然后调整解码序列的大小和重排序。由于需要完整的重新编码操作，需要复杂的帧重排序和完整尺度的运动估计。因而，人们开展了许多研究工作来降低复杂度负担同时保持编码性能［C5，C24］。下面讨论 H.264 和 VC-1 之间的转码。假定所考虑的档次限于 H.264 的基本档次和 VC-1 的简单档次，编码参数诸如帧内/帧间宏块模式、运动矢量、参考帧和跳过宏块需要被分析和相互比较。

8.7.1　帧内编码宏块模式映射

当 H.264 比特流中的一个帧内编码宏块编码为一个 VC-1 帧内编码宏块时，H.264 帧内编码宏块可被编码为内部的 4×4（9 个不同方向模式）块或内部 16×16（4 个不同模式）块，但一个 VC-1 帧内编码宏块有四个 8×8 块且没有预测模式。既然 VC-1 的帧内编码宏块使用 8×8 变换，那么我们不考虑 H.264 的块大小（16×16 或 4×4），就不必保持 H.264 帧内预测类型信息。表 8.13 给出了提出的帧内编码宏块映射方案。

表 8.13　H.264 和 VC-1 帧内编码宏块映射方案［C29］

H.264 帧内编码宏块	VC-1 帧内编码宏块
帧内编码的 16×16 宏块（任意模式）	帧内编码的 8×8 宏块
帧内编码的 4×4 宏块（任意模式）	帧内编码的 8×8 宏块

8.7.2 帧间编码宏块模式映射方案

H.264 的帧间编码宏块有 7 种不同的运动补偿块大小——16×16，16×8，8×16，8×8，4×8，8×4，4×4（见图 8.26），VC-1 的帧间编码宏块只有 2 种不同的运动补偿块大小——16×16 和 8×8。另一个明显的差别是 H.264 使用 4×4（在保真度范围扩展这的 8×8）的变换大小，而 VC-1 使用 4 种变换大小——8×8，4×8，8×4，4×4。通常，H.264 所选的 16×16 运动补偿块是针对相对均匀的区域，不会映射到 VC-1 中以 8×8 为变换大小的 16×16 宏块。运动补偿块 8×16，16×8 具有小的不均匀运动，因而它们映射到 VC-1 中的帧间编码的 8×8 块，因为 16×16 宏块的大小会由于不均匀运动产生较差的质量。由于使用所选的 H.264 块大小作为块同源性的度量，就可决定 VC-1 的变换大小并加以应用。换言之，H.264 块大小决定了这个特定块所用的变换大小。这个方法不需要计算确定变换大小的每个 8×8块的半和值与半差值。

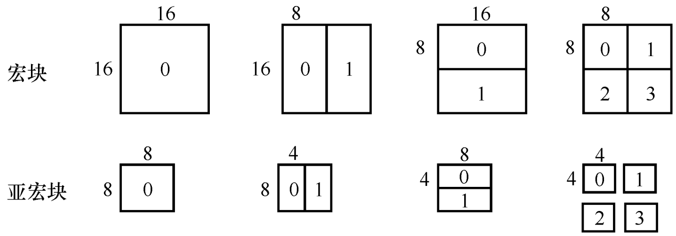


图 8.26 自适应 ME/MC 预测（七个块大小）的宏块和亚宏块分割，带有运动矢量的编码块按光栅扫描顺序进行排序（2010 IEEE）

通常 H.264 对于具有不均匀运动的区域选择 8×8，8×4，8×8 和 4×4 模式。对于这样的不均匀宏块，VC-1 去除了 16×16 模式。因为 H.264 的块大小决定了 VC-1 要用的变换大小，那么宏块就映射到 VC-1 中 8×8 大小的块。表 8.14 描述了用在 VC-1 中的帧间编码宏块和变换类型的映射决策。

表 8.14 H.264 和 VC-1 帧间编码宏块映射与 VC-1 变换类型［C29］

H.264 帧间编码宏块	VC-1 帧间编码宏块	VC-1 变换大小
帧间编码 16×16	帧间编码 16×16	8×8
帧间编码 16×8	帧间编码 8×8	8×4
帧间编码 8×16	帧间编码 8×8	4×8
帧间编码 8×8	帧间编码 8×8	8×8
帧间编码 4×8	帧间编码 8×8	4×8
帧间编码 8×4	帧间编码 8×8	8×4
帧间编码 4×4	帧间编码 8×8	4×4

8.7.3　运动矢量映射

重复使用 H.264 用过的运动矢量能显著降低 VC-1 编码的复杂度。因为两个标准中的变换编码块大小不总是对等映射，那么当 H.264 宏块具有多于一个运动矢量的时候就可以选择运动矢量中值［C29］。表 8.15 描述了运动矢量的选择情况。除了帧间编码 16×16 和帧间编码 8×8 模式的情况，有必要从若干已有的运动矢量中为每个 8×8 块选择一个运动矢量。其他选择运动矢量代表的组合也是可能的。

表 8.15　H.264 与 VC-1 帧间编码宏块运动矢量映射［C29］

H.264 帧间编码宏块	VC-1 帧间编码宏块	运动矢量重用决策
帧间编码 16×16	帧间编码 16×16	对 16×16 块，取相同的运动矢量
帧间编码 16×8	帧间编码 8×8	对每个 8×8 块，取平均运动矢量
帧间编码 8×16	帧间编码 8×8	对每个 8×8 块，取运动矢量的均值
帧间编码 8×8	帧间编码 8×8	对每个 8×8 块，取相同的运动矢量
帧间编码 4×8	帧间编码 8×8	对每个 8×8 块，取运动矢量的中值
帧间编码 8×4	帧间编码 8×8	对每个 8×8 块，取运动矢量的中值
帧间编码 4×4	帧间编码 8×8	对每个 8×8 块，取运动矢量的中值

8.7.4　参考帧

H.264/AVC 标准对于运动估计定义了使用多达 16 个参考帧，而 VC-1 按照分片类型 P 或 B 分别只用一个或两个参考帧。运动矢量的重用意味着使用相同的参考帧来保持它们的意义。如果我们假定 H.264 基本档次（单参考帧）和 VC-1 的简单档次（单参考帧），那么可以使用作为输入比特流的同样的参考帧，且不需要缩放运动矢量。

8.7.5　跳过宏块

当在比特流中指出了一个跳过宏块，那么就不会发送那个宏块更多的数据。H.264 中跳过宏块到 VC-1 的跳过宏块的模式转换是一个直接的前向过程。因为两个标准对于跳过宏块的定义是完全兼容的，所以直接转换是可以的。

8.8　VC-1 的传输

VC-1 的基本流（ES，Elementary Stream）可被封装并在 MPEG-2 标准的传输流和程序流中指出［S3］。传输流（TS，Transport Stream）规定了一个容器格式封装包化基流（PES，Packetized Elementary Streams），并具有纠错和流同步化特征以

维护信号退化时传输的完整性。程序流（PS，Program Stream）是一个容器格式，用于复用数字音频、视频和高可靠度媒体如光盘。

基本流由访问单元组成（AU，Access Units），该单元包括所有的帧编码数据和填充比特，后面接续下一个 AU 的起始码。编码数据单元表示一个视频帧，不管该帧被编为渐进模式还是隔行渐进模式，隔行帧隔行模式还是隔行场隔行模式。如果该帧不是由一个序列起始码、序列首部或入口点首部开头的话，那么 AU 就以帧起始码开头。否则，AU 以帧起始码之前的这些结构（排除任何填充的字节）的第一个字节作为开头。一个 AU 也包括随便哪一个帧或场级的用户数据起始码和用户数据字节。

基本流本身不适合在信道上传输，因为它没有为解码器和显示提供任何定时信息。没有定时信息，出现一个随机访问的时候，解码器可能会与其他的基本流发生冲突。因此，通常情况是首先将其包化成或者是定长包或者是变长包来形成包化基本流。VC-1 的 PES 在 MPEG-2 的系统框架中生成。VC-1 简单档次/主档次的 PES 包净负荷格式首部插入在 MPEG-2 PES 中，如图 8.27 所示。当 PES 包净负荷包括 VC-1 基本流数据字节时，PES 包首部的域诸如 PES 包长度、流 ID、数据对齐指示、播放时间戳（PTS，Presentation Time Stamp）、解码时间戳（DTS，Decoding Time Stamp）和间断指示，就被修改或增强。PES 包长度是一个 16bit 的域，规定了字节数。VC-1 基本流的流 ID 设置为 0xFD，表明使用了 ISO 13818-1，修正案 2 的扩展机制。如果它设置为 1，数据对齐指示用来表明 PES 包首部紧跟这视频起始码或音频同步字。如果数据对齐指示设置为 1，且没有节目映射表（PMT，Program Map Table）中 VC-1 基本流相关的数据对齐亚描述符，那么默认的对齐类型值应该等于 0x02。数据对齐指示值‘0’表明没有规定对齐。对于简单档次或主档次 VC-1 基本流，数据对齐指示域的值应该总是设置为 1 且没有与 VC-1 基本流相关的数据对齐亚描述符。

PTS/DTS 戳按照与 MPEG-2 完全相同的方式使用。特别地，PTS/DTS 域的值与第一个视频 AU 相关，AU 开始于 PES 包的净负荷中。VC-1 基流的域 stream_id_extension 具有范围在 0x55 和 0x5F 之间的任何值。stream_id 和 stream_id_extension 的组合明确地定义了携带 VC-1 视频数据 PES 包。

discontinuity_indicator 是个 1 比特的域，表明当前传输流包是否处于不连续状态。连续性计数器在传输流包中使 VC-1 的数据发生中断之后，具有相同包标识的传输流包中基本流数据首字节就是 VC-1 访问点或后接访问点的 VC-1 序列端起始码的首字节。VC-1 访问点定义如下：

- 如果序列首部之前没有序列起始码，那么 VC-1 序列首部的首字节可以是一个访问点。
- 如果序列起始码就在序列首部之前，那么序列起始码的首字节就是一个访问点。

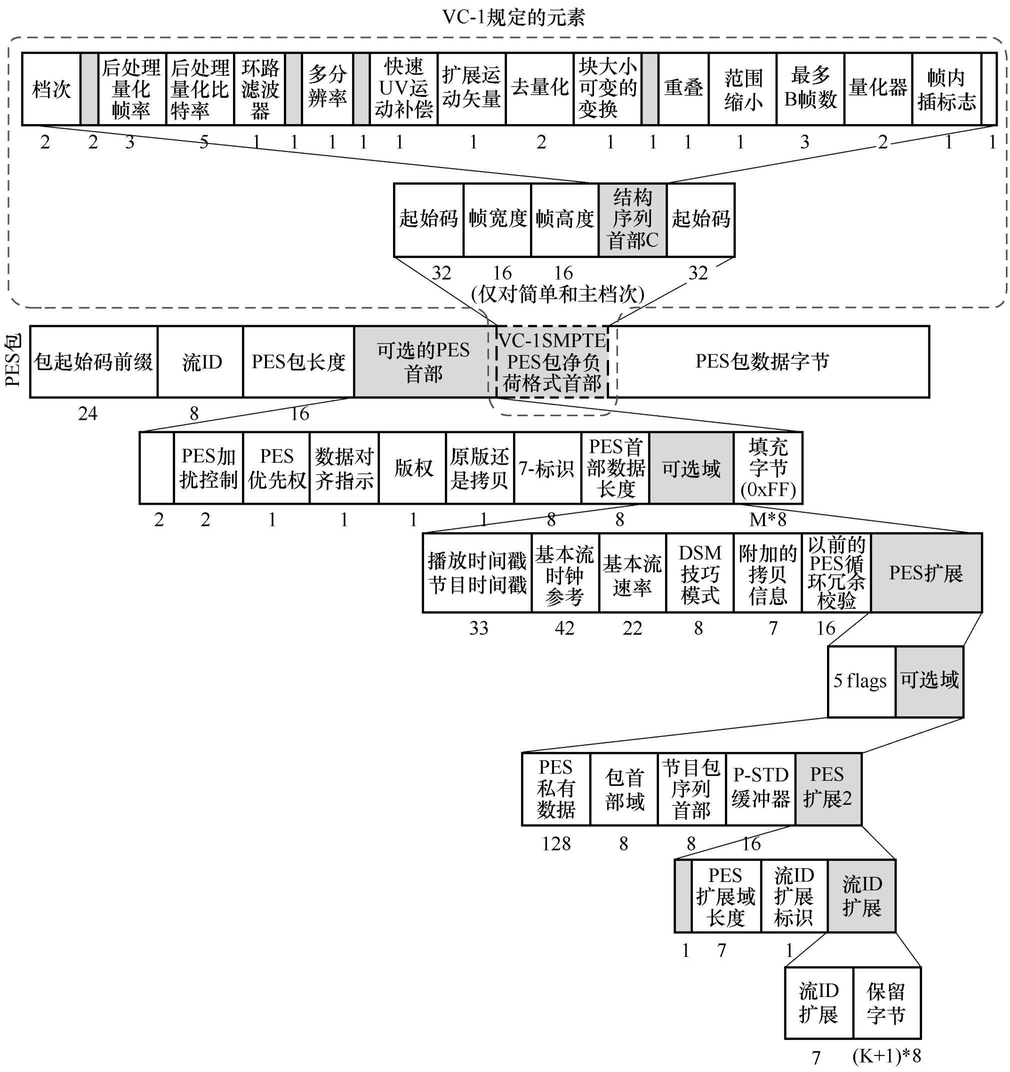


图 8.27　VC-1 基本流的 PES 包语法框图［C12］（2007 SMPTE）

8.8.1　传输流中 VC-1 的数据封装

在传输流中，节目（原文：programs——译者注）用列表集合指出，这些列表周期性地发出，叫作节目规定信息（PSI，Program Specific Information）。特别低，节目映射表（PMT，Program Map Table）提供了节目细节并规定了必要的信息诸如包标识（PID，Packet Identifier）来找到和编码基本流分量。MPEG-2 传输流中的 VC-1 基本流的传送取决于 MPEG-2 系统规定中的传输系统目标解码器（T-STD，Transport System Target Decoder）缓冲模型。

首先，PMT 的 stream_type 值设置为 0xEA 来表明 VC-1 基本流。这意味着基本流被当作是一个私自管理的流。这个私有值的范围必须由注册描述符来捕获，它包括若干子描述符。MPEF-2 系统中的注册描述符设计的目的是唯一且明确地识别“私有”数据的格式。privately-managed（）的结构最初视为 MPEG-2 系统设计的，存在于 MPEG-2 程序单元（PE，Program Element）的内部描述符中，PE 在相应于 VC-1 基本流的传输流节目映射（PM：program map）部分。Subdescriptor_tag 定义了档次/级别、队列、缓冲大小和静止帧子描述符，如图 8.28 所示。

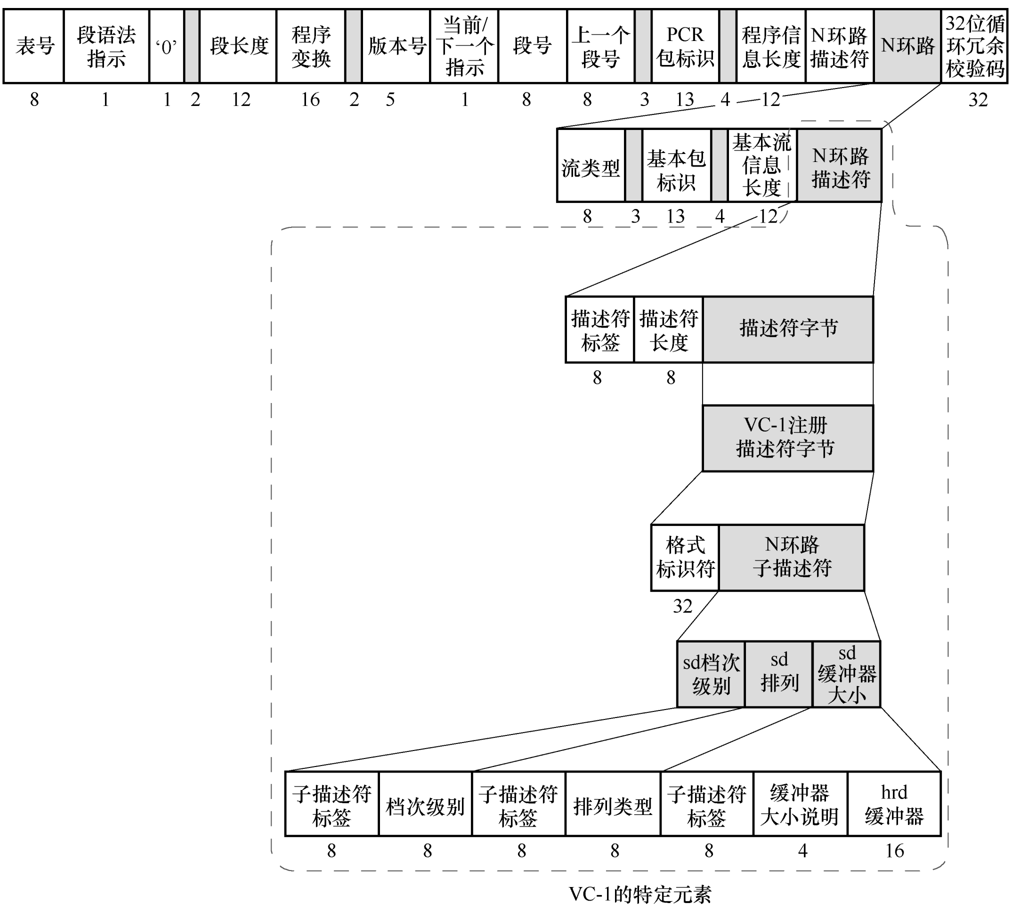


图 8.28　VC-1 基本流的传输流程序映射段图［C12］（2007 SMPTE）

8.8.2　程序流中的 VC-1 数据封装

MPEG-2 程序流中 VC-1 基本流的发送取决于 MPEG-2 系统规定的程序系统目标解码器（P-STD，Program System Target Decoder）缓冲模型。流类型值 0xEA 和

注册描述符和前面部分定义的子描述符还可用于 MPEG-2 程序流中 VC-1 基本流的输送。唯一的区别是，在 MPEG-2 程序流的情况下，这些域所处的结构由程序流映射（PSM，Program Stream Map）定义，PSM 包括若干基本流，而传输流中的封装由程序映射表（PMT，Program Map Table）来管理，PMT 只与 MPEG-2 的传输流相关。这样，图 8.29 所示的封装图看起来类似于传输流的封装图，除了基本流的 N- 环路之外。

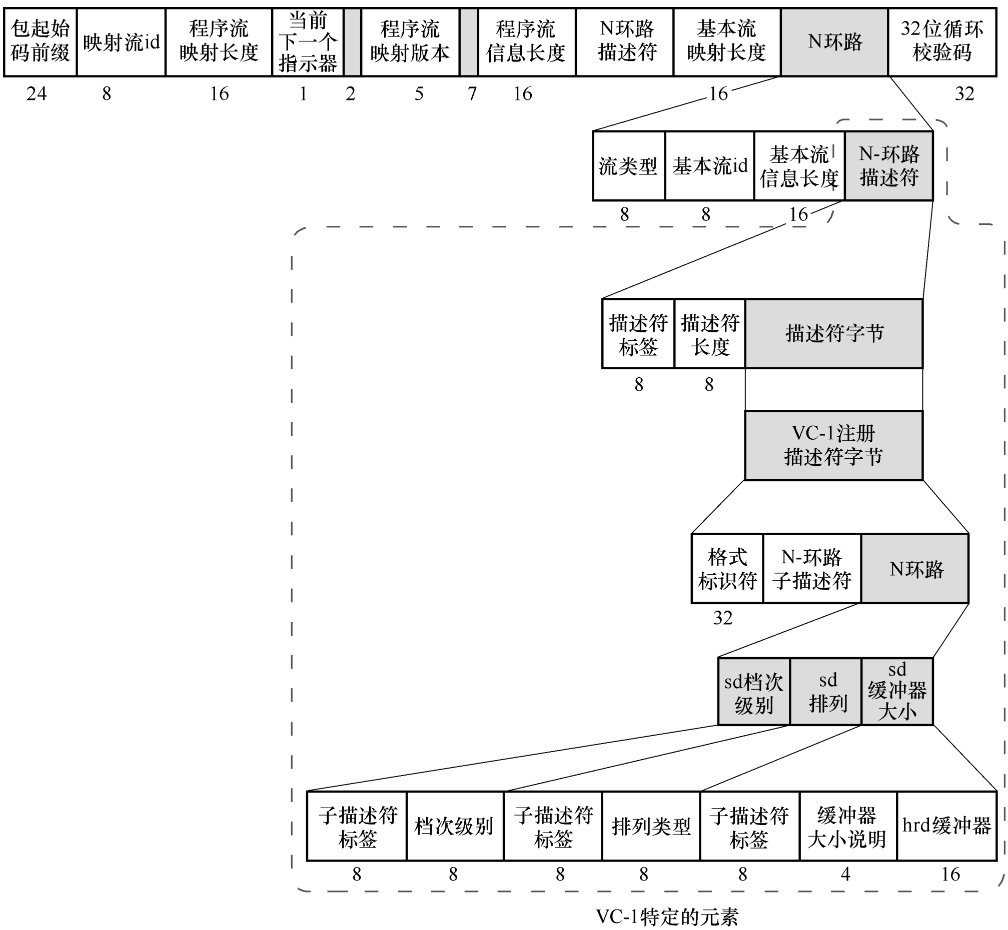


图 8.29　VC-1 基本流程序流映射图［C12］(2007 SMPTE)

8.9　VC-2 视频压缩

SMPTE 标准（SMPTE ST 2042-1：2012）

下面有关 VC-2［C6］的观点是从上文引用的 SMPTE 标准复制的。

8.9.1 前言

这一节提供了大量信息，却没有形成这篇文档的一个完整部分。VC-2 标准规定了一个视频压缩系统的压缩流语法和参考解码器的操作。VC-2 是个帧内编码的视频压缩系统，服务于专业应用，它在许多分辨率上提供了有效的编码，包括各种不同类型的 CIF、标清电视和高清电视。VC-2 利用小波变换，将视频信号分解成多个频带。该编译码器设计得简单又灵活，能够在很宽范围的分辨率和应用领域上进行工作。

该系统提供如下能力：

- 多分辨率变换：用小波变换对数据进行编码，数据包逐子带进入比特流中。高压缩比导致分辨率的逐渐损失。较低分辨率输出的图像可以通过仅仅抽取较低分辨率的数据来获得。
- 帧和场编码：帧和场都可被单独编码。
- 恒定比特率和可变比特率：VC-2 既允许恒定比特率又允许可变比特率操作。对于低时延图像，对于图像的每个区域（VC-2 分片）比特率都是恒定的，以此来确保恒定的时延。
- 可变位深：8、10、12、和 16 位格式及以上都支持。
- 多色差采样格式：4:4:4、4:2:2 和 4:2:0 视频都支持。
- 无损和 RGB 编码：有一个公共工具箱既可用于有损编码又可用于无损编码。RGB 编码或者由 YCoCg 整数颜色变换［H23］提供支持以获得最大压缩效率，或者直接压缩 RGB 信号。
- 小波滤波器：一系列小波滤波器可被用于平衡性能和复杂度。支持 Daubechies（9，7）滤波器以与 JPEG2000 兼容。提供了保真度滤波器实现改进的分辨率可扩展性。
- 简单的流导航：编码流形成了一个双向列表，每个帧首部都指出了到前一帧和下一帧的偏移量，这样就能支持场精度的高速浏览而不需要分析或解码。
- 多档次：VC-2 提供了多个档次以解决特定应用的特殊需求。为了最好地匹配它们的目标应用，不同的档次包括或省略了特定的编码工具。主档次用核心语法提供了最大的压缩效率、可被比特率编码和无损编码。简单档次通过使用简单的变长码而不是主档次所用的算术编码进行熵编码，提供了较低复杂度的编译码，但压缩效率也较低。针对需要很低的、固定时延的应用，低时延档次使用了一个修改的语法。这可能是像输入或输出视频的几行一样低的时延。面向低带宽设施的再次使用，低时延档次适合于轻微压缩，例如在 SD-SDI 线路上传送高清电视信号。高质量档次类似地提供了低时延的轻微压缩，且支持可变比特率和无损编码。

8.9.2　范围

这个标准从流语法、熵编码、系数析出过程和帧解码过程定义了 VC-2［C6］视频压缩。解码器的操作通过混合伪代码和数学运算进行定义。

VC-2 是一个帧内编码的视频编译码器，使用小波变换以及熵编码，可以容易地以很高的比特率在硬件或软件上实现。另外的标准和推荐的练习可能对于特定的应用对编码定义特殊的限制。

VC-2 编码器和解码器的功能框图分别如图 8.30 和图 8.31 所示。

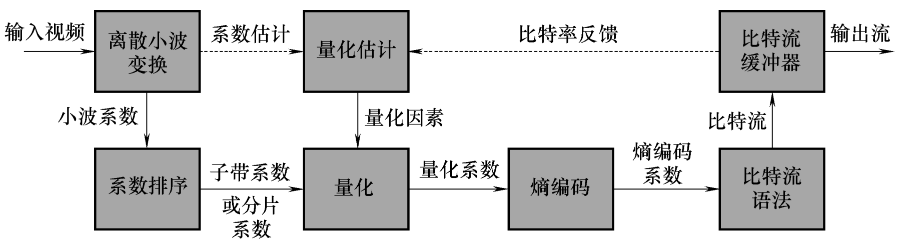


图 8.30　样值编码器功能框图［C6］（SMPTE 2012）

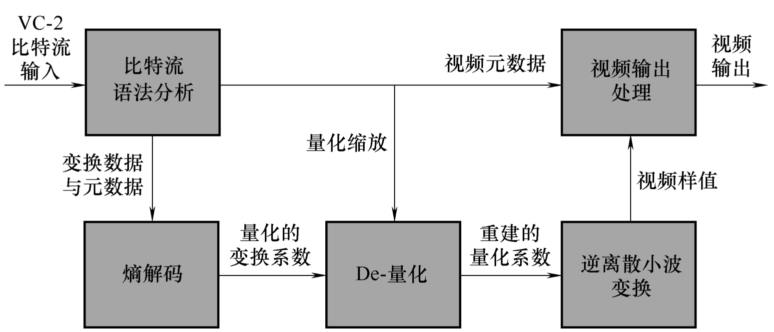


图 8.31　VC-2 解码器的功能框图（SMPTE 2012）

8.10　总结

本章既讨论了基于微软 WMV9 的 VC-1，又讨论了基于 BBC 开发的 DIRAC 帧内编码的 VC-2。VC-1 和 VC-2 都是 SMPTE 的标准。

8.11　专题研究

P. 8.1　Vijayakuar［C29］开发了一个低复杂度的 H.264 到 VC-1 的转码器。使用各种测试序列（CIF 和 QCIF），VC-编译码器在低量化参数（QP）下的主

观质量方面可比得上 H.264 编译码器。请实现这个转码器并证实［C29］中的结论。将这个转码器与级联型转码器在不同的 QP 下进行对比。

P.8.2　Vijayakuma［C29］提出的转码器仅仅涉及 H.264 的基本档次［H7］和 VC-1 的简单档次［C3］。请将该转码器扩展到其他档次诸如主档次和高档次。注意，这个专题可以作为硕士水平的研究基础。

P.8.3　Vijakumar［C29］在她的硕士论文（6.5 节的结论和未来的工作）中提出，H.264 到 VC-1 的转码器可以通过运动矢量（MV）细化实现进一步的改善。请找到一个适当的 MV 细化方案来支持所提出的 MV 重用思想，并对于不同档次在不同比特率下用各种视频序列（CIF、QCIF 等等）实现 H.264 到 VC-1 的转码器。请将该转码器与级联转码器进行对比。

附言：硕士论文［C29］可以从 www-ee.uta.edu/dip 下载。点击 courses 再点击 EE5359。向下滚动并查看列表“List of recent theses/projects”。点击 Vijakumar thesis。议案、ppt 等也能被下载。

P.8.4　Lee 和 Kalva［C5］开发了一个 VC-1 到 H.264 的高效转码器，请实现之。

附录 A　关于 Dirac、H. 264 和 H. 265 图像质量的调查研究

本附录基于 Biju Shrestha［E44］的专题。

欲知详情，请参见 UTA 的 EE5359 课程网址 http://www-ee.uta.edu/Dip/Courses/EE5359/index.html 上的专题（2012 春季）。

A. 1　前言

Shrestha［E44］针对 CIF 序列实现了 H. 265，并与 Dirac 和 H. 264 进行了对比。有几个视频压缩标准与它们的旧版本［H46］相比都具有性能和质量方面的额外改进。Dirac、H. 264 和 H. 265 的图像质量可以使用一些指标诸如 PSNR、CSNR、MSE、SSIM、MSSIM 和 FSIM［Q13、Q27、Q28］用各种测试序列加以研究。传统的 PSNR 和 MSE 是一种亮度测量的指标，不能用于测量主观质量［Q16］。像 SSIM 和 FSIM 这样的指标考虑了人类的视觉系统。

A. 2　H. 265

H. 265 也叫作 HEVC［E5］，它能表现出相对于 AVC（ITU-T H. 264 | ISO/IEC

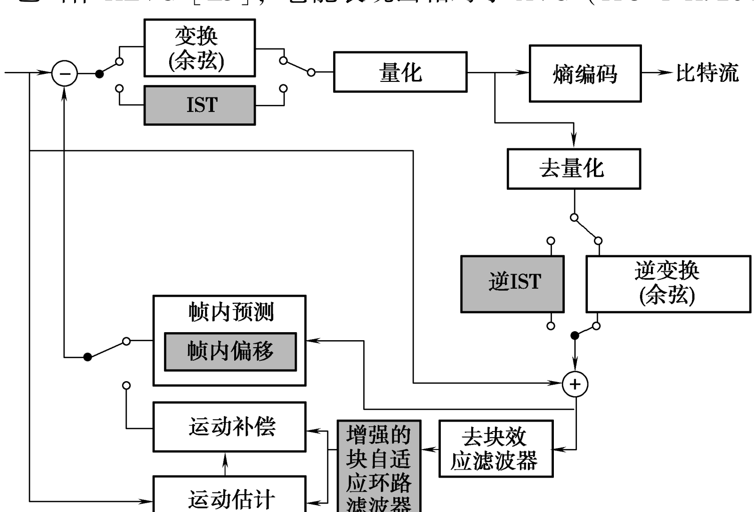


图 A. 1　H. 265 的编码器框图，灰色框是所提出的工具，白色框是 H. 264/AVC 工具［E10］（2011 ETRI）

14496-10）［E5］明显改善的压缩性能。Alshina 等［E2］研究了高分辨率高清 1080p 下的编码效率并得出结论，与 MPEG-4 AVC［E2］相比，对于分级的 B 结构和 IPPP 结构能提高平均 37% 和 36% 的比特节约。通常的基于块的视频编译码器由许多环节组成，包括帧内预测和帧间预测、变换、量化、熵编码和滤波［E10］，如图 A. 1 和图 A. 2 所示。在过去的十年中，视频编码技术经历了深入研究以其获得更高的编码效率。

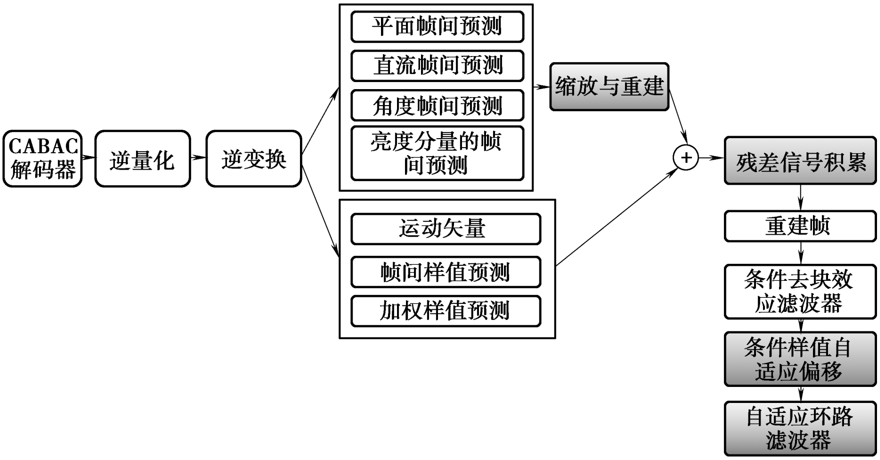


图 A. 2　H. 265 的解码器框图。灰色框是提出的工具，白色框是 H. 264/AVC 的工具［E10］（2011 ETRI）

A. 3　使用 SSIM 和 FSIM 的图像质量评价

在不同的阶段，诸如采集、处理、压缩、存储、传输和再生的过程中，数字图像和视频容易产生各种失真［Q2］。这种恶化导致了视觉质量的下降。由几个广泛用于定量描述图像质量的指标，像 FSIM、SSIM、比特流、PSNR 和 MSE［Q28，Q2，Q13，Q17］。传统指标像 PSNR 和 MSE 直接依赖图像的强度，与主观保真度级别没有关系［Q16］。MSE 不能非常精确地把人类视觉系统模型化［Q19］。度量参数如 PSNR、MSE、Dirac 的 SSIM、H. 264 和 H. 265 将被进行比较，以研究它们相对特性并给出结论。

SSIM 是基于结构信息恶化的图像质量评价方法［Q13］。SSIM 采取了一种方法，便于人类视觉系统适应于从图像抽取结构信息［Q17］。因而，保持结构信号对于图像保真度测量是很重要的。图 A. 3 给出了非结构和结构失真的差别。非结构失真是一些参数的改变，如亮度、对比度、伽玛失真和空间移位，通常由环境和仪器条件造成，发生在图像采集和显示过程中［Q17］。在另一方面，结构失真包括加性噪声、抖动和有损压缩［Q17］。结构失真改变了图像的结构［Q17］。图

A. 4 解释了用在计算 SSIM 的度量系统。

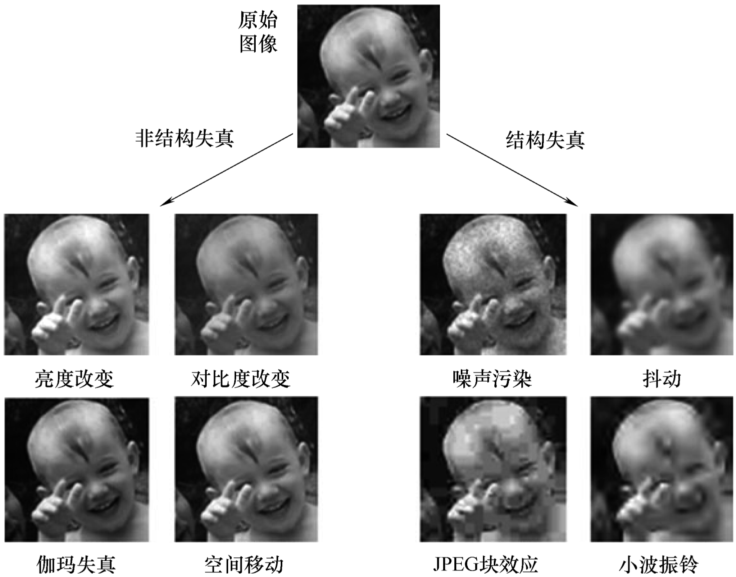


图 A. 3　非结构失真与结构失真的差别［Q22］(2009 IEEE)

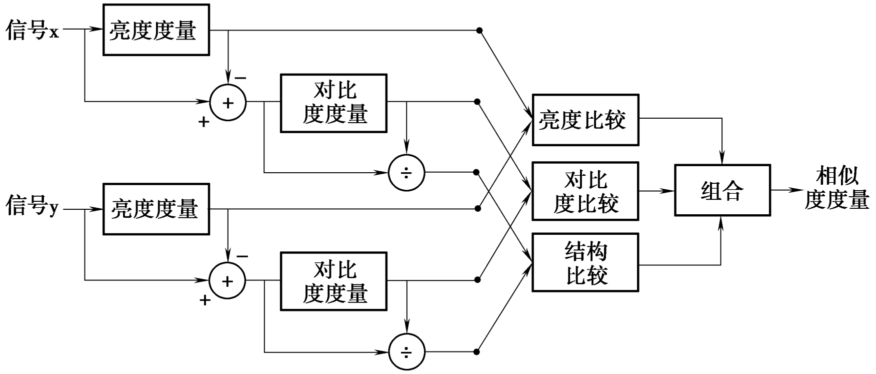


图 A. 4　SSIM 度量系统框图［Q22］(2009 IEEE)

对于给定的矢量 $\underline{x} = \{x_i \mid i = 1, 2, \cdots, N\}$ 和 $\underline{y} = \{y_i \mid i = 1, 2, \cdots, N\}$，SSIM 在三种不同的指标上进行计算，如亮度、对比度和结构，结构由式（A. 1），式（A. 2）和式（A. 3）分别进行数学描述［Q17］。

$$l(\underline{x},\underline{y}) = \frac{2\mu_{\underline{x}}\mu_{\underline{y}} + C_1}{\mu_{\underline{x}}^2 + \mu_{\underline{y}}^2 + C_1} \tag{A. 1}$$

$$c(\underline{x},\underline{y}) = \frac{2\sigma_{\underline{x}}\sigma_{\underline{y}} + C_2}{\sigma_{\underline{x}}^2 + \sigma_{\underline{y}}^2 + C_2} \tag{A. 2}$$

$$s(\underline{x},\underline{y})=\frac{\sigma_{\underline{xy}}+C_3}{\sigma_{\underline{x}}\sigma_{\underline{y}}+C_3} \tag{A.3}$$

式中，$\mu_{\underline{x}}$和$\mu_{\underline{y}}$分别等于$\underline{x}$和$\underline{y}$的局部样值的平均；$\sigma_{\underline{x}}$和$\sigma_{\underline{y}}$分别等于$\underline{x}$和$\underline{y}$的局部样值的标准差；$\sigma_{\underline{xy}}$等于$\underline{x}$和$\underline{y}$的局部样值的相关系数；C_1、C_2和C_3等于常数，它在分母变小时可以稳定计算。

SSIM 指标的一般形式可以通过综合式（A.1）、式（A.2）和式（A.3）[Q27]来得到。

$$\mathrm{SSIM}(\underline{x},\underline{y})=[l(\underline{x},\underline{y})]^{\alpha}[c(\underline{x},\underline{y})]^{\beta}[s(\underline{x},\underline{y})]^{\gamma} \tag{A.4}$$

式中，α、β和γ是调整这三个分量相对重要性的参数。如果$\alpha=\beta=\gamma=1$，我们得到[Q27]

$$\mathrm{SSIM}(\underline{x},\underline{y})=\frac{(2\mu_{\underline{x}}\mu_{\underline{y}}+C_1)(2\sigma_{\underline{xy}}+C_2)}{(\mu_{\underline{x}}^2+\mu_{\underline{y}}^2+C_1)(\sigma_{\underline{x}}^2\sigma_{\underline{y}}^2+C_1)} \tag{A.5}$$

图 A.5 给出了用 MSE 和 SSIM 定量描述的不同失真的图像。可以很清楚地看到，不同的图像具有基于人类视觉系统（HVS，Human Visual System）的不同质量。然而，所有失真图像具有近似相同的 MSE，然而对于低质量图像 SSIM 是比较小的，比 MSE 给出好得多的图像质量指示。

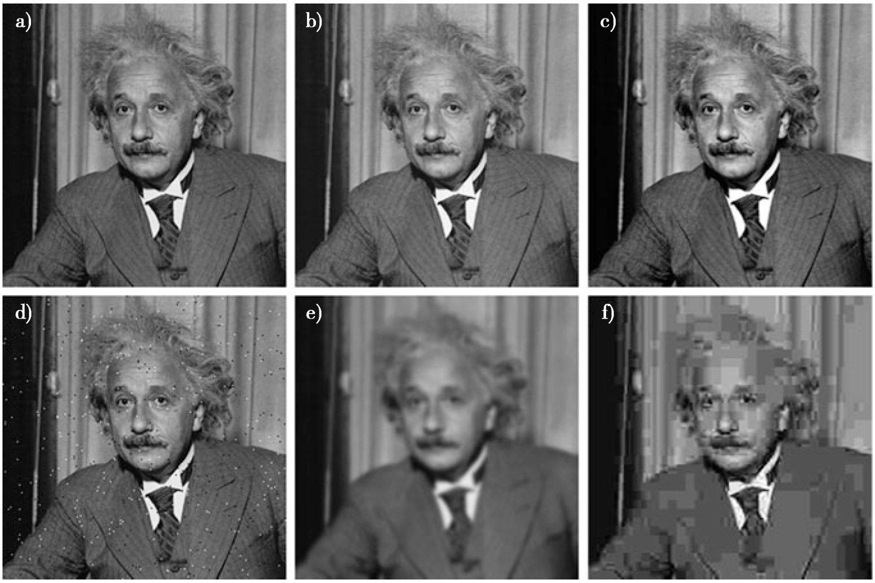


图 A.5　不同失真下图像的 MSE 和 SSIM 度量值

a）原始图像（MSE =0，SSIM =1）　b）平均亮度平移（MSE =144，SSIM =0.988）　c）对比度拉伸（MSE =144，SSIM =0.913）　d）脉冲噪声污染（MSE =144，SSIM =0.840）　e）抖动（MSE =144，SSIM =0.694）　f）JPEG [J18] 压缩（MSE =142，SSIM =0.662）[Q13]

特征相似度指标基于这样的事实，HVS 主要按照图像的低级特征来理解一幅图像［Q28］。相位一致性（PC，Phase Congrency）是一种对于局部结构重要性的无量纲的测度［Q28］。PC 和图像梯度幅值（GM，Gradient Magnitude）量度在 FSIM 中分别用作主要和次要特征［Q28］。通过把 PC 作为图像局部质量的加权函数可以算出 FSIM 分值，而局部质量由 PC 和 GM 表征［Q28］。FSIM 是面向灰度级图像而设计的［Q28］，而 FSIMc 混合了色度信息。FSIM 可以数学建模如下［Q28］

$$\mathrm{FSIM}(\underline{x},\underline{y})=\frac{\sum_{x\in\Omega}S_{\mathrm{L}}(\underline{x})\mathrm{PC}_{\mathrm{m}}(\underline{x})}{\sum_{x\in\Omega}\mathrm{PC}_{\mathrm{m}}(\underline{x})} \tag{A.6}$$

式中，$S_{\mathrm{L}}(\underline{x})$ 等于参考图像和失真图像的总体相似度。

FSIM 可以数学建模如式（A. 7）所示，计算过程如图 A. 6 所描述［Q28］。

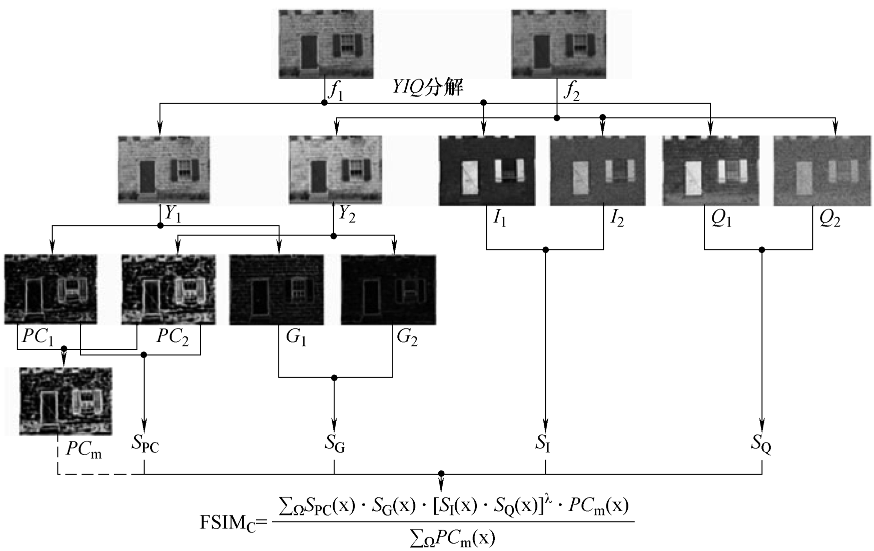


图 A. 6　FSIM/FSIM$_{\mathrm{c}}$ 指标计算图示

（f_1是参考图像，f_2是f_1的失真版本）［Q28］（2011　IEEE）

$$\mathrm{FSIM_C}=\frac{\sum_{x\in\Omega}S_{\mathrm{L}}(\underline{x})[S_{\mathrm{C}}(\underline{x})]^{\lambda}\mathrm{PC}_{\mathrm{m}}(\underline{x})}{\sum_{x\in\Omega}\mathrm{PC}_{\mathrm{m}}(\underline{x})} \tag{A.7}$$

式中，$\lambda>0$ 是用来调整色度分量重要性的参数。

$$\mathrm{FSIM_C}=\frac{\sum_{x\in\Omega}S_{\mathrm{PC}}(\underline{x})S_{\mathrm{G}}(\underline{x})[S_{\mathrm{I}}(\underline{x})S_{\mathrm{Q}}(\underline{x})]^{\lambda}\mathrm{PC}_{\mathrm{m}}(\underline{x})}{\sum_{x\in\Omega}\mathrm{PC}_{\mathrm{m}}(\underline{x})}$$

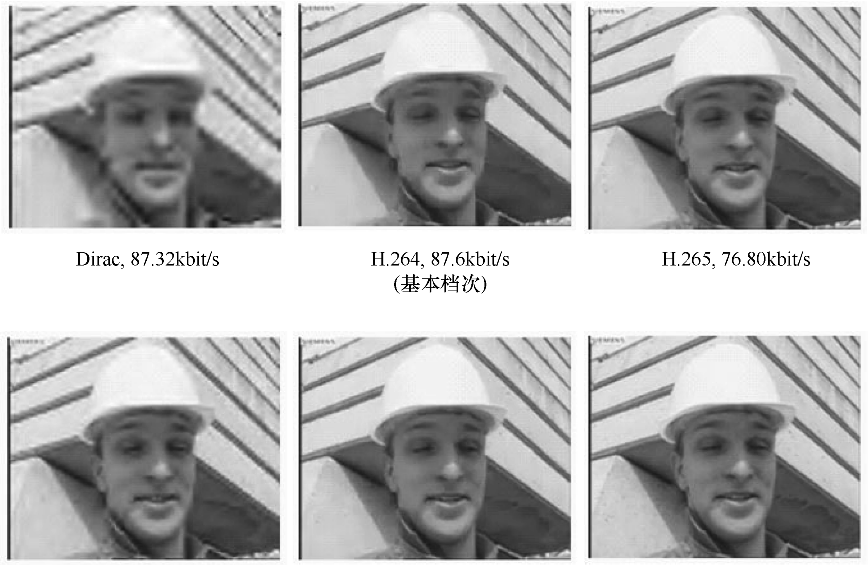
所有的指标都使用不同的方法来定量比较图像。这种方法使得方法之间彼此不同。表 A. 1 给出了图像质量评价指标性能在六个数据块上的排序。从表 A. 1 可以看出，在进行图像质量评价时，FSIM 比 SSIM 好，SSIM 比 PSNR 好。

表 A.1 图像质量评价指标性能排序（FSIM、SSIM 和 PSNR）[**Q28**]

	TID2008	CSIQ	LIVE	IVC	MICT	A57
FSIM	1	1	1	1	1	1
SSIM	2	2	2	2	2	2
PSNR	3	3	3	3	3	3



图 A.7 原始的 Foreman QCIF 视频序列 [V3]
（视频信息——QCIF：QCIF 序列：foreman_qcif. yuv；帧高 176；
帧宽 144；帧率 30 帧/s；用于编码的帧总数：30 帧）



Dirac, 87.32kbit/s

H.264, 87.6kbit/s
(基本档次)

H.265, 76.80kbit/s

Dirac, 152.85kbit/s

H.264, 142.82kbit/s
(基本档次)

H.265, 162.46kbit/s

图 A.8 用不同的编译码器 QCIF 序列 Foreman 的结果 [E44]



Dirac, 397.60kbit/s　　H.264, 323.76kbit/s (基本档次)　　H.265, 398.21kbit/s



Dirac, 4266.92kbit/s　　H.264, 3667.01kbit/s (基本档次)　　H.265, 2301.14kbit/s

图 A. 8　用不同的编译码器 QCIF 序列 Foreman 的结果［E44］（续）

A. 4　结果

A. 4. 1　使用 QCIF 序列 Foreman 的结果

图 A. 9 ~ 图 A. 11 和表 A. 2。

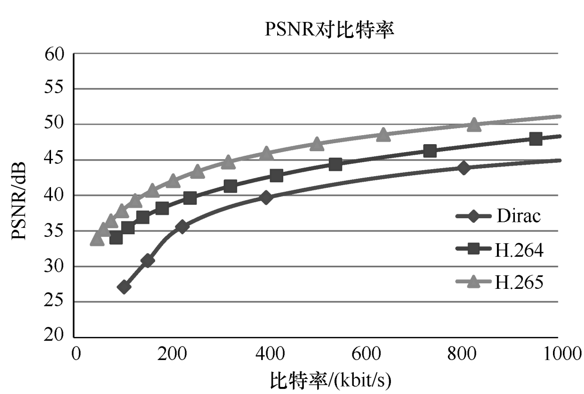


图 A. 9　对于 QCIF 序列 Foreman 在各种比特率下获得的 PSNR［E44］

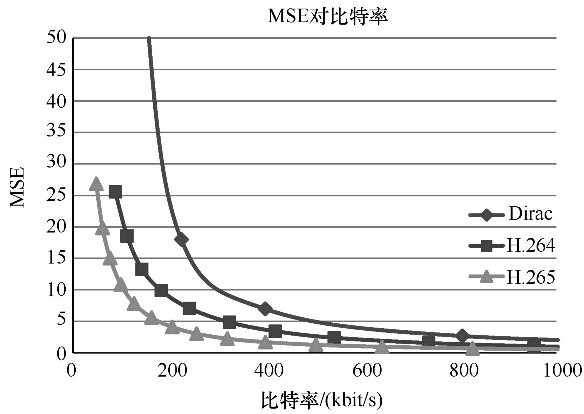


图 A. 10 对 QCIF 序列 Foreman 在各种比特率下实现的 MSE [E44]

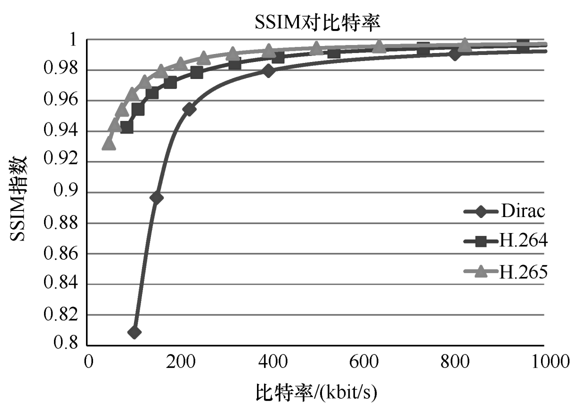


图 A. 11 对于 QCIF 序列 Foreman 在各种比特率下实现的 SSIM [E44]

表 A. 2 QCIF 序列 Foreman 的 Y 分量的结果表格 [E44]

Dirac			
比特率/(kbit/s)	PSNR/dB	MSE	SSIM
87. 31543	24. 2585	243. 9108	0. 70942
104. 1455	27. 10295	126. 7024	0. 80875
152. 8516	30. 81619	53. 88419	0. 89656
224. 4629	35. 58351	17. 9775	0. 95434
397. 6055	39. 69524	6. 97519	0. 97947
805. 6953	43. 87075	2. 66689	0. 99047
1615. 6	47. 5301	1. 14834	0. 99547
2918. 021	50. 96503	0. 52069	0. 99801
4266. 925	55. 04958	0. 20329	0. 99934

（续）

H. 264			
比特率/(kbit/s)	PSNR/dB	MSE	SSIM
3667. 01	69. 941	0. 00723	1
3280. 58	63. 357	0. 0309	0. 9999
2903. 03	58. 825	0. 08601	0. 9996
2352. 62	55. 135	0. 20001	0. 9991
1952. 06	53. 245	0. 30894	0. 9987
1586. 51	51. 563	0. 45541	0. 9981
1205. 7	49. 597	0. 71567	0. 9971
954. 55	47. 961	1. 04312	0. 9959
736. 1	46. 25	1. 54575	0. 9942
540. 58	44. 37	2. 38023	0. 9917
418. 68	42. 799	3. 41684	0. 9885
323. 76	41. 293	4. 83252	0. 9845
240. 47	39. 632	7. 08134	0. 9784
182. 57	38. 186	9. 88001	0. 972
142. 82	36. 904	13. 27354	0. 9652
111. 9	35. 448	18. 55763	0. 9544
87. 6	34. 056	25. 56888	0. 9426
H. 265			
比特率/(kbit/s)	PSNR/dB	MSE	SSIM
2301. 136	61. 0449	0. 09232	0. 99946
1810. 152	56. 4716	0. 2055	0. 99881
1407. 104	53. 6141	0. 34358	0. 99812
1080. 7	51. 5846	0. 51131	0. 99738
827. 016	49. 9778	0. 71189	0. 99655
639. 616	48. 5619	0. 96235	0. 99556
502. 432	47. 2557	1. 27745	0. 99439
398. 208	45. 9397	1. 70959	0. 99277
319. 312	44. 7036	2. 25273	0. 99081
255. 792	43. 3888	3. 03526	0. 98797

（续）

H. 265			
比特率/(kbit/s)	PSNR/dB	MSE	SSIM
205. 376	42. 0663	4. 09816	0. 98408
162. 456	40. 7085	5. 59949	0. 97936
126. 6	39. 2593	7. 81681	0. 97236
99. 016	37. 8324	10. 84688	0. 9643
76. 808	36. 416	15. 03856	0. 95421
61. 152	35. 2166	19. 84122	0. 94431
48. 152	33. 8996	26. 82359	0. 93215



图 A. 12　原始 CIF 序列 Foreman［V3］

（视频信息——CIF 序列：Foreman_cif. yuv；帧高：352；帧宽：288；帧率：30 帧/s；用于编码的总帧数：30 帧）



图 A. 13　使用不同的编译码器 CIF 序列 Foreman 的测试结果［E44］



Dirac, 251.79kbit/s



H.264, 96.64kbit/s
(基本档次)



H.265, 93.44kbit/s



Dirac, 295.83kbit/s



H.264, 320.87kbit/s
(基本档次)



H.265, 274.82kbit/s

图 A. 13 使用不同的编译码器 CIF 序列 Foreman 的测试结果［E44］（续）

A. 4. 2 使用 CIF 序列 Foreman 的结果

图 A. 14 ~ 图 A. 18 和表 A. 3。

A. 4. 3 使用 QCIF 序列 Container 的结果

图 A. 19 ~ 图 A. 23 和表 A. 4。

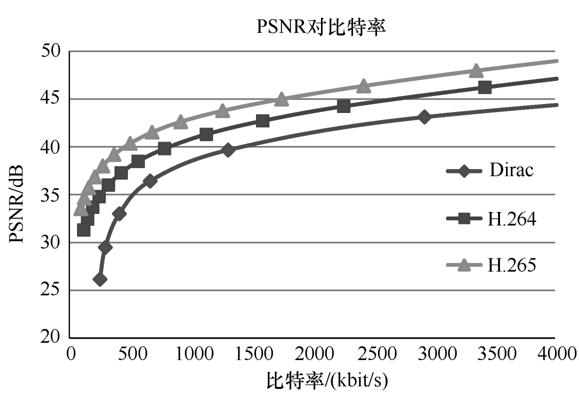


图 A. 14 CIF 序列 Foreman 在各种比特率下实现的 PSNR［E44］

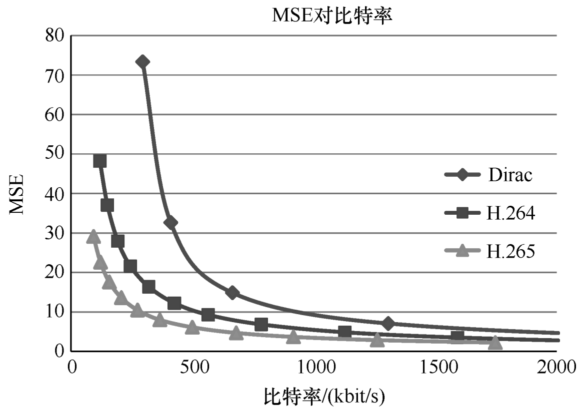


图 A. 15　CIF 序列 Foreman 在各种比特率下的 MSE［E44］

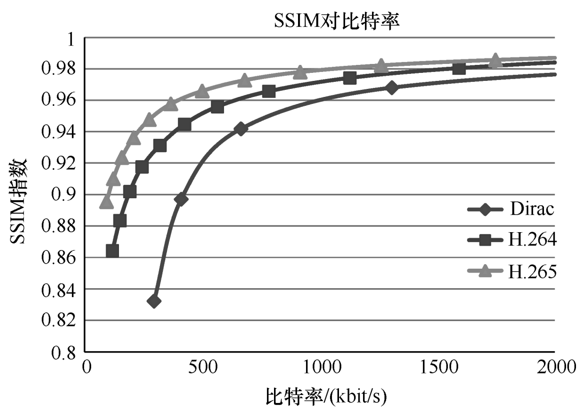


图 A. 16　CIF 序列 Foreman 在各种比特率下的 SSIM［E44］

表 A. 3　CIF 序列 Foreman 的 Y 分量测试结果列表

Dirac			
比特率/(kbit/s)	PSNR/dB	MSE	SSIM
251. 7861	26. 16213	157. 3501	0. 75122
295. 8262	29. 17884	73. 31557	0. 83221
411. 4258	32. 99665	32. 61482	0. 89702
665. 3672	36. 42028	14. 82696	0. 94178
1307. 283	39. 65805	7. 03519	0. 96798
2926. 172	43. 10598	3. 1804	0. 98192
6706. 156	46. 82782	1. 34989	0. 99189

（续）

Dirac			
比特率/(kbit/s)	PSNR/dB	MSE	SSIM
12275. 79	50. 82581	0. 53765	0. 99706
17673. 92	55. 18275	0. 19715	0. 99912
H. 264			
比特率/(kbit/s)	PSNR/dB	MSE	SSIM
14905. 66	64. 495	0. 02357	0. 9999
13361. 46	59. 419	0. 0748	0. 9995
11138. 92	55. 569	0. 18081	0. 9988
9462. 26	53. 504	0. 29052	0. 9981
7787. 24	51. 79	0. 43092	0. 9972
5876. 26	49. 722	0. 69383	0. 9956
4650. 4	48. 045	1. 02124	0. 9937
3423. 99	46. 211	1. 56024	0. 9904
2259. 58	44. 232	2. 45871	0. 9856
1592. 58	42. 713	3. 48797	0. 9804
1128. 79	41. 309	4. 8165	0. 9742
784. 31	39. 805	6. 80952	0. 9657
565. 55	38. 478	9. 2431	0. 9559
426. 02	37. 27	12. 21212	0. 9448
320. 87	36. 003	16. 34605	0. 9313
244. 86	34. 799	21. 57591	0. 9176
193. 04	33. 681	27. 91477	0. 9019
150. 36	32. 449	37. 0522	0. 8834
118. 94	31. 307	48. 19565	0. 8642
96. 64	30. 181	62. 47732	0. 8413
H. 265			
比特率/(kbit/s)	PSNR/dB	MSE	SSIM
9562. 624	57. 5297	0. 15687	0. 99874
7740. 52	54. 2904	0. 28891	0. 9977
6062. 28	51. 8845	0. 47837	0. 99625
4588. 064	49. 8117	0. 7684	0. 99415
3351. 448	47. 9604	1. 17528	0. 99149
2424. 128	46. 3638	1. 6593	0. 98867

（续）

H. 265			
比特率/(kbit/s)	PSNR/dB	MSE	SSIM
1748. 224	44. 9876	2. 20962	0. 98562
1262. 032	43. 7665	2. 85814	0. 98215
916. 832	42. 618	3. 66935	0. 97789
681. 096	41. 5195	4. 67979	0. 9727
500. 16	40. 3466	6. 10686	0. 96587
366. 656	39. 171	7. 982	0. 95762
274. 816	37. 9924	10. 45239	0. 94766
208. 016	36. 847	13. 59112	0. 9361
158. 512	35. 7304	17. 55409	0. 92362
122. 136	34. 6362	22. 57885	0. 91007
93. 44	33. 5346	29. 07615	0. 89551



图 A. 17　原始的 QCIF 序列 container ［V3］

（视频信息——QCIF 序列：container_qcif. yuv；帧高 176；帧宽 144；帧率：30 帧/s；用于编码的总帧数：30 帧）



Dirac, 47.08kbit/s　　H.264, 38.54kbit/s (基本档次)　　H.265, 44.81kbit/s

图 A. 18　使用不同编译码器 QCIF 序列 container 的测试结果［E44］



Dirac, 74.55kbit/s



H.264, 69.37kbit/s
(基本档次)



H.265, 74.51kbit/s



Dirac, 218.86kbit/s



H.264, 203.5kbit/s
(基本档次)



H.265, 205.55kbit/s



Dirac, 3568.48kbit/s



H.264, 1806.49kbit/s
(基本档次)



H.265, 1743.176kbit/s

图 A. 18　使用不同编译码器 QCIF 序列 container 的测试结果［E44］（续）

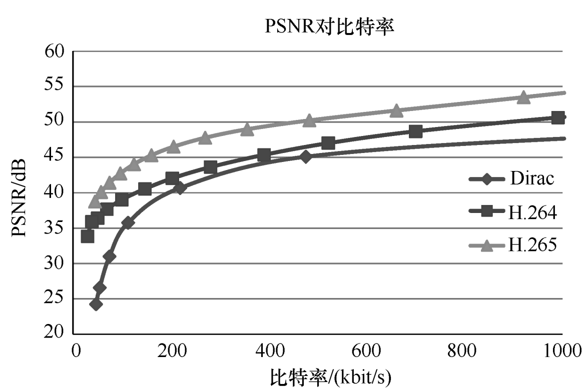


图 A. 19　QCIF 序列 container 在各种比特率下的 PSNR［E44］

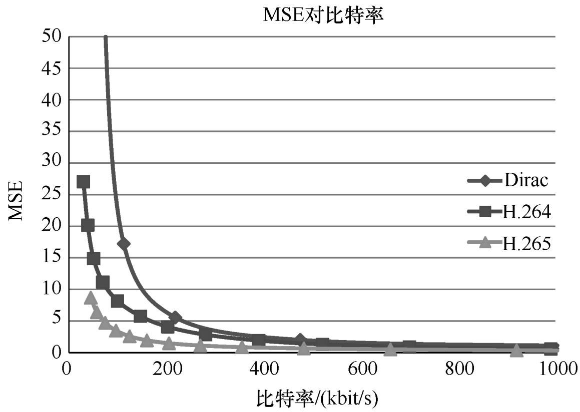


图 A. 20　QCIF 序列 container 在各种比特率下的 MSE［E44］

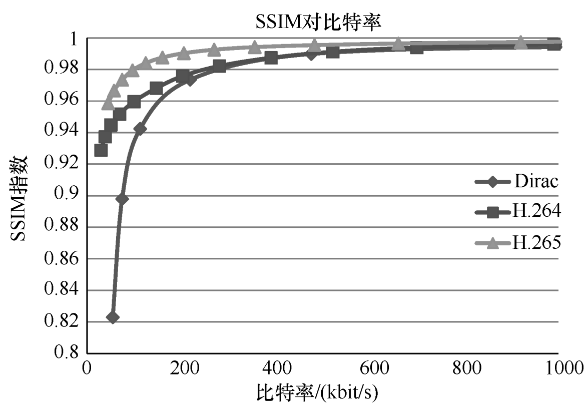


图 A. 21　QCIF 序列 container 在各种比特率下的 SSIM［E44］

表 A. 4　QCIF 序列 container 的 Y 分量测试结果列表

Dirac			
比特率/(kbit/s)	PSNR/dB	MSE	SSIM
47. 08594	24. 25652	244. 0219	0. 74807
54. 68457	26. 58447	142. 769	0. 82296
74. 55176	30. 9816	51. 87055	0. 8979
112. 7266	35. 77031	17. 22063	0. 94235
218. 8604	40. 67826	5. 5623	0. 97352
475. 7578	45. 09811	2. 01034	0. 98987

（续）

Dirac			
比特率/(kbit/s)	PSNR/dB	MSE	SSIM
1121. 212	48. 0381	1. 02158	0. 99476
2325. 415	50. 91785	0. 52638	0. 99743
H. 264			
比特率/(kbit/s)	PSNR/dB	MSE	SSIM
3095. 22	68. 477	0. 01008	0. 9999
2737. 65	62. 433	0. 03835	0. 9998
2371. 05	58. 482	0. 09337	0. 9994
1806. 49	54. 779	0. 21739	0. 9986
1350. 69	52. 611	0. 3576	0. 9976
990. 99	50. 622	0. 56554	0. 9961
700. 1	48. 674	0. 88545	0. 9939
521. 56	47	1. 3013	0. 9912
390. 56	45. 359	1. 89787	0. 9874
280. 98	43. 626	2. 82744	0. 9822
203. 05	42. 038	4. 07523	0. 9758
146. 91	40. 541	5. 75502	0. 9681
99. 92	39. 024	8. 15939	0. 9597
69. 37	37. 661	11. 17344	0. 9517
50. 52	36. 418	14. 87515	0. 9448
38. 54	35. 93	20. 16539	0. 9373
29. 63	33. 82	27. 03133	0. 9288
H. 265			
比特率/(kbit/s)	PSNR/dB	MSE	SSIM
1743. 176	60. 7041	0. 10723	0. 99917
1267. 496	56. 1859	0. 22991	0. 99822
920. 84	53. 4967	0. 35728	0. 99736
661. 048	51. 6196	0. 50069	0. 99645
482. 976	50. 2206	0. 65918	0. 99543
355. 896	48. 977	0. 85664	0. 99419
269. 928	47. 7722	1. 11536	0. 99254
205. 552	46. 514	1. 47679	0. 99031
160. 176	45. 3014	1. 94302	0. 98755

（续）

H. 265			
比特率/(kbit/s)	PSNR/dB	MSE	SSIM
124. 56	44. 0425	2. 58736	0. 98402
96. 608	42. 7314	3. 49207	0. 97945
74. 512	41. 4227	4. 71294	0. 97346
57. 36	40. 0978	6. 3907	0. 96656
44. 808	38. 757	8. 69522	0. 95846

A. 4. 4 使用 CIF 序列 Container 的结果

图 A. 24 ~ 图 A. 27 和表 A. 5。



图 A. 22 原始 CIF 序列 container ［V3］

（视频信息——CIF 序列：container_cif. yuv；帧高：352；帧宽：288；帧率：30 帧/s；用于编码的总帧数：30 帧）



Dirac, 122.01kbit/s　　H.264, 115.82kbit/s (基本档次)　　H.265, 122.14kbit/s

图 A. 23 使用不同编译码器时 CIF 序列 container 的测试结果 ［E44］





Dirac, 217.54kbit/s　　H.264, 233.19kbit/s (基本档次)　　H.265, 208.02kbit/s





Dirac, 906.96kbit/s　　H.264, 753.17kbit/s (基本档次)　　H.265, 916.83kbit/s





Dirac, 15501.95kbit/s　　H.264, 1555.43kbit/s (基本档次)　　H.265, 11469.4kbit/s

图 A. 23　使用不同编译码器时 CIF 序列 container 的测试结果［E44］（续）

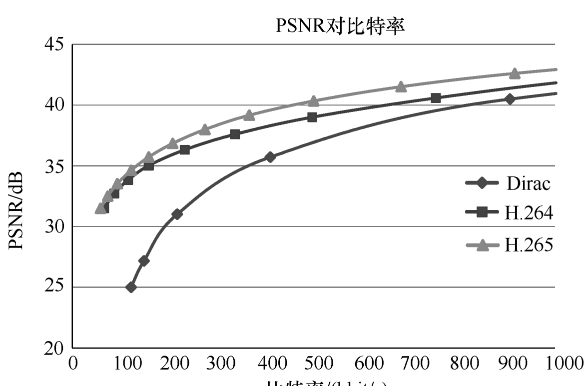


图 A. 24　CIF 序列 container 在各种比特率下的 PSNR［E44］

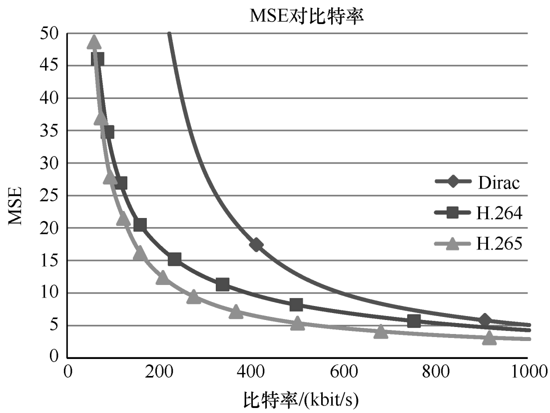


图 A. 25　CIF 序列 container 在各种比特率下的 MSE ［E44］

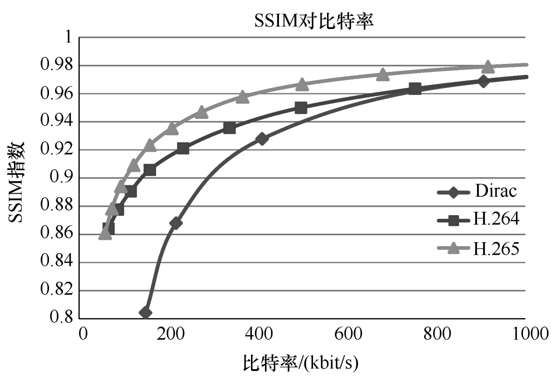


图 A. 26　CIF 序列 container 在各种比特率下的 SSIM ［E44］

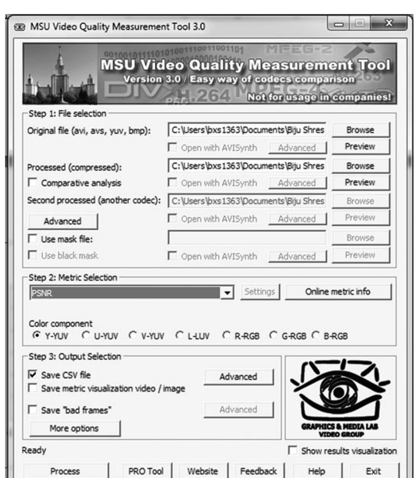


图 A. 27　MSU 视频质量评价工具 3. 0。免费版本计算不同的指标如 PSNR、MSE 和 SSIM ［Q30］

表 A. 5　CIF 序列 container 的 Y 分量测试结果列表

Dirac			
比特率/(kbit/s)	PSNR/dB	MSE	SSIM
122. 0107	24. 99744	205. 7484	0. 73993
148. 9033	27. 16859	124. 802	0. 80432
217. 5371	31. 00858	51. 54929	0. 86798
410. 5264	35. 71859	17. 42695	0. 92785
906. 9629	40. 49885	5. 7969	0. 96866
2139. 141	44. 28241	2. 42571	0. 98573
5243. 777	47. 30256	1. 21011	0. 99267
10287. 48	50. 76457	0. 54528	0. 9971
15501. 95	55. 16019	0. 19818	0. 9991
H. 264			
比特率/(kbit/s)	PSNR/dB	MSE	SSIM
16414. 2	71. 623	0. 00592	1
12895. 66	64. 052	0. 02622	0. 9998
11403. 13	59. 113	0. 08037	0. 9994
9215. 22	55. 314	0. 19186	0. 9986
7593. 62	53. 349	0. 3012	0. 9977
5989. 81	51. 676	0. 44254	0. 9967
4355. 95	49. 564	0. 71983	0. 9947
3288. 64	47. 736	1. 09712	0. 9922
2316. 39	45. 758	1. 73043	0. 9879
1555. 43	43. 834	2. 6944	0. 9819
1076. 42	42. 151	3. 96953	0. 9739
753. 17	40. 588	5. 68885	0. 9635
497. 22	39. 006	8. 18687	0. 95
337. 19	37. 597	11. 32449	0. 9355
233. 19	36. 32	15. 1952	0. 9211
158. 5	35. 017	20. 51233	0. 9059
115. 82	33. 836	26. 92392	0. 8906
87. 29	32. 723	34. 78167	0. 8776
65. 95	31. 503	46. 04602	0. 864
H. 265			
比特率/(kbit/s)	PSNR/dB	MSE	SSIM
11469. 4	62. 3897	0. 0796	0. 99938

（续）

H. 265			
比特率/（kbit/s）	PSNR/dB	MSE	SSIM
9562. 624	57. 5297	0. 17563	0. 99861
7740. 52	54. 2904	0. 30634	0. 99756
6062. 28	51. 8845	0. 48965	0. 9961
4588. 064	49. 8117	0. 76738	0. 99396
3351. 448	47. 9604	1. 09722	0. 99164
2424. 128	46. 3638	1. 43027	0. 9896
1748. 224	44. 9876	1. 85426	0. 98697
1262. 032	43. 7665	2. 41575	0. 98339
916. 832	42. 618	3. 12276	0. 97917
681. 096	41. 5195	4. 08571	0. 97363
500. 16	40. 3466	5. 36587	0. 9664
366. 656	39. 171	7. 15771	0. 95772
274. 816	37. 9924	9. 46237	0. 947
208. 016	36. 847	12. 41334	0. 9354
158. 512	35. 7304	16. 22794	0. 92336
122. 136	34. 6362	21. 48924	0. 90917
93. 44	35. 5346	27. 86858	0. 89417
73. 52	32. 508	36. 96402	0. 87824
58. 424	31. 5021	48. 6575	0. 86083

A. 5 结论

本附录旨在研究不同视频编译码器的质量性能，主要集中于 Dirac、H. 264 和 H. 265［D24，H36，E50］。对于所有的三个视频编译码器，计算了各种比特率下的不同参数如 PSNR，MSE 和 SSIM，进行了对比研究。基于不同空间/时间分辨率的各种测试序列，广泛地使用了 MATLAB、Microsoft Visual Studio、和 MSU 视频质量评价工具（见图 A. 27）［Q30］在各种比特率下对不同的编译码器进行了图像质量评价。图 A. 9 ~ 图 A. 11、图 A. 14 ~ 图 A. 16、图 A. 19 ~ 图 A. 21、图 A. 24 ~ 图 A. 26分别给出了 QCIF 序列 Forman、CIF 序列 Foreman、QCIF 序列 container 和 CIF 序列 container 分别在各种比特率下指标如 PSNR，MSE，和 SSIM 的变化。表格形式的结果在表 A. 2 ~ 表 A. 5 中给出。基于所获得的结果，H. 265 在任何给定的比特率上都实现了超过 H. 264 和 Dirac 的性能。然而，就所用的测量指标而言，在较

高的比特率上，所有的三个编译码器性能彼此接近。

A. 6　专题研究

P. A. 1　在本附录中，利用各种测试序列（QCIF 和 CIF）在不同的比特率上基于 SSIM、PSNR 和 MSE 比较了 H. 264、H. 265 和 Dirac。请实现这些测试并证实相应结果。

P. A. 2　请基于 MS SSIM 和 FSIM 推广 P. A. 1 中的比较。

P. A. 3　请将 P. A. 1 中的比较推广到标清电视和高清电视序列。

附录 B　AVSNR 软件的 PSNR 平均值

本附录讨论的是关于 JVT 专家组邮件清单上的三个颜色分量的平均 PSNR［1］。

（JVT 使用了一些由亚琛工业大学主办的电子邮件清单，主要是 jvt-experts@lists. rwth-aachen. de——这些列表可用 https://mailman. rwth-aachen. de 加入）。

我在用 4∶2∶2 的高档次进行测试。我的工作是比较位深为 10bit/像素与 8bit/像素的客观评价指标，但对于色度分量利用 4∶2∶2 的亚采样。我们应该用什么样的平均 PSNR 来使用 AVSNR 软件呢？

事实上，没有一个大家都很满意的方法。在 JCTVC-H0012［2］中有关于这些问题的一些讨论。

在某种程度上，用比特分配效应有可能“愚弄评价指标”，例如，像 JCTVC-G0401［3］描述的那样。

在某种程度上，我仍然认为有必要分别看一下三个颜色分量的每一个 PSNR 的值。

但是，寻找一种简单办法得出一个组合度量衡以便于综合评价，这是非常取的。

组合指标在三个颜色分量中公平地表示率失真的相对平衡，这是令人满意的。对于 4∶2∶0 视频，通常亮度质量关于比特花费的增额等于色度质量的某种增额，该亮度增额是很不均匀的。对于每一份亮度的 PSNR 损失量，你能获得多少色度的 PSNR 增加量之间，可能会有大约 10∶1 的比率——我认为它是一个大于 3∶1 的比率并由下式反映

$$PSNR_{avg} = (6 \times PSNR_Y + 1 \times PSNR_U + 1 \times PSNR_V)/8 \qquad (B.1)$$

然而，组合指标表示视觉质量的均衡也是可取的。我认为，视觉质量均衡不像 PSNR 均衡那样极端。

参考文献

［1］G. Sullivan，［jvt-experts］PSNR avrage for avsnr software，May16，2012.

［2］http://phnix. it-sudparis. eu/jct/doc_end_user/currnt_document. php？ id =5039

［3］http://phnix. it-sudparis. edu/jct/doc_end_user/current_document. php？ id =3658

附录 C　通用图像质量指标与 SSIM 的比较

本附录基于 Chaitanya Chukka［Q29］的专题。

欲知详情，请参见 UTA 的 EE5359 课程网址 http://www-ee.uta.edu/Dip/Courses/EE5359/index.html 上的项目（2010 春季）。

C.1　前言

Caitanya Chukka 比较了一个通用的图像质量指标和 SSIM［Q29］。在此附录中，研究了一个新的通用客观图像质量指标，它可以用于各种图像处理应用当中。这一易于计算的指标把任一种失真模型化为三个因素的组合：相关度损失、亮度失真和对比度失真。本专题的目的是用这个指标来确定图像质量，而且判定结果好于传统的误差和方法，诸如均方误差（MSE，Mean Square Error）、峰值信噪比（PSNR，Peak Signal to Noise Ratio）、方均根误差（RMSE，Root Mean Square Error）和评价绝对误差（MAE，Mean Absolute Error）。然而，在此专题中，我们只对比我们在数学上定义的质量因数和图像的 MSE 来定义一幅好的图像。这种方法不依赖于测试图像的类型和大小。它也独立于图像的像素大小和观察条件。这样，“通用”一词用于这种方法就是恰如其分的。Q 的动态范围从 1 到 -1，正 1 比指标为负 1 表示一幅更好的图像。

另外，还研究了基于 HVS 假设的结构相似测度，假设人类视觉感知高度自适应于抽取场景的结构信息。这个测度是一个可自由选用的框架，用于基于结构信息恶化的质量评价。被对象表面的亮度是照度和反射率的乘积，但场景中对象的结构与照度无关。图像的结构信息定义为赋予场景对象结构的东西，它独立于平均亮度和对比度。

图像信号通常是静止的，随空间而变化的；它的失真来自于多方面的侵害，如脉冲椒盐噪声、加性高斯噪声、乘性斑点噪声、对比度拉伸、抖动和 JPEG 压缩。首先用 MATLAB 代码对 Lena 图像计算 MSE、PSNR、Q、和 SSIM 中的每一个，然后再对其他测试图像进行计算（Couple 和 Goldhill）。最后给出结果得出结论。

C.1.1　MSE:

MSE（Mean Square Error，均方误差）是个信号保真度测度。信号保真度测度的目的是通过提供一个定量分数来比较两个信号，这个分数描述了相似度/保真度，或者反过来，就是它们之间的误差/失真水平。通常，假定其中一个信号是未受影响的原始信号，而另一个受到误差导致的失真或污染。

假定 $x = \{x_i \mid i=1, 2, \cdots N\}$ 和 $y = \{y_i \mid i=1, 2, \cdots N\}$ 是两个定长的离散

信号（例如，视觉图像），其中 N 是信号样值数（像素，如果信号是图像），x_i 和 y_i 分别是 x 和 y 的第 i 个样值。信号 x 和 y 之间的 MSE 是

$$\mathrm{MSE}(x,y) = \frac{1}{N}\sum_{i=1}^{N}(x_i - y_i)^2 \tag{C.1}$$

在 MSE 中，我们会经常提到误差信号 $e_i = x_i - y_i$，它是原始信号与失真信号之间的差值。如果其中一个信号是质量可接受的（或者未受损伤的）原始信号，另一个是评价其质量的失真信号，那么 MSE 还可被视为一种信号质量的测度。

更一般的形式是 l_p 范数

$$d_p(x,y) = \left(\sum_{i=1}^{N}|e_i|^p\right)^{1/p} \tag{C.2}$$

MSE 经常转换成峰值信噪比（PSNR）测度

$$\mathrm{PSNR} = 10\log_{10}\frac{L^2}{\mathrm{MSE}} \tag{C.3}$$

式中，L 是可允许的图像像素强度动态范围。例如，对于分配 8bit/灰度级的图像，$L = 2^8 - 1 = 255$。如果进行比较的图像具有不同的动态范围，那么 PSNR 是很有用的，不过此外并没有比 MSE 包含更多新的信息。

C.1.2　为什么用 MSE？［Q22］

MSE 具有许多引人注意的特征：

1）简单。它没有参数，计算容易，对每个样值只要一次乘法和两次加法的复杂度。另外，需要较少的内存——平方误差可以在每个样值上计算，与其他样值无关。

2）物理意义明确——它是定义误差能量很自然的方法。任何正交（或单一的）线性变换之后，这样的能量测度都可保持，例如傅里叶变换（Parseval 定理）。这种能量保持特性保证了变换域的信号失真能量与信号域的能量相同。

3）在优化场合下 MSE 是一种优越的度量指标。因为 MSE 的梯度矩阵和 Hessian 矩阵容易计算，最小 MSE（MMSE）优化问题常常具有封闭形式的解析解；如果做不到这一点，迭代的数值优化程序也常常是容易设计出来的。

4）MSE 广泛使用，只因为它是一个惯例。历史上，它被广泛用于优化和评价各种信号处理应用，包括滤波器设计、信号压缩、复原、去噪、重建和分类。而且，遍观文献，性能相当的算法大多经常用 MSE/PSNR 进行比较。因而，它提供了一个方便且宽泛的标准，新算法的 MSE/PSNR 可以与之对比。这节约了时间和精力但进一步推广了 MSE 的使用。

C.1.3　MSE 有什么问题？［Q22］

很显然，MSE 拥有许多对于应用和分析令人满意的特性，但读者可能会指出忽略了一个更为基础的问题。那就是，MSE 真的度量了信号的保真度？面对它的

迷人特征，一个信号处理从业者可能会选择 MSE，如果它被证明是一种合理的信号保真度指标。但情况真是如此吗？很不幸，当 MSE 用于预测人类对于图像保真度和质量感知的时候，相反的情况出现了。图 C.1［Q22］给出了一个说明图例，其中一幅爱因斯坦图像用不同类型的失真进行了修改：对比度拉伸、亮度均值平移、加性高斯白噪声污染、脉冲噪声失真、JPEG 压缩［Q16］［JP7］［JP8］、抖动、空间缩放、空间平移和旋转。

在图 C.1 中，给出了 MSE 值和另一个质量指标的值，结构相似度（SSIM）值。SSIM 指标将在后面详细描述。注意到，若干失真图像的 MSE 值（相对于原始图像 C.1a）几乎是相同的（见图 C.1b～C.1g），即使同样的图像呈现出明显不同的视觉质量。我们还注意到经过轻微的几何修改的图像（见图 C.1h～图 C.1i）相对于原始图像可能会有很大的 MSE 值，然而只表现出一点可忽略的视觉质量的损失。所以，一个自然的问题是：“MSE 有什么问题吗？”［Q22］。



图 C.1　对于不同类型失真改变的“爱因斯坦”图像保真度测度的比较［Q22］（2009　IEEE）
a）参考图像　b）平均对比度拉伸　c）亮度平移　d）高斯噪声污染　e）脉冲噪声污染
f）JPEG 压缩［JP9］　g）抖动　h）空间缩放（缩小）　i）空间平移（向右）
j）空间平移（向左）　k）旋转（逆时针）　l）旋转（顺时针）

C.1.4 使用 MSE 的隐含假定

1）原始图像样值之间的信号保真度独立于时间或空间关系。换言之，如果原始信号和失真信号以相同的方式随机重排序，那么它们之间的 MSE 不会改变。

2）信号保真度独立于原始信号与误差信号之间的任何关系。对一个给定的误差信号，MSE 保持不变，不管它加到哪个原始信号上。

3）信号保真度独立于误差信号样值的符号。

4）所有的信号样值对于信号保真度同等重要。

不幸的是，在度量图像保真度视觉感知方面，它们没有一个（即使粗略地）能成立。关于这些假定的诚实性，文献［Q22］的图 3 展示了给人印象深刻的 MSE 的失败的图例。

C.1.5 主观对客观图像质量测度

既然人类是多数图像图像处理应用中最终的接受者，那么评价一幅图像质量的最可靠方法就是主观评价。事实上，评价意见得分（MOS，Mean Opinion Score），一个需要若干人类观察者服务的主观质量测度，很久以来被认为是图像质量度量的最好方法。然而，MOS 方法代价不菲，而且在现实应用中太慢而显得没用。

客观图像质量评价研究的目的是设计计算模型，使其能够准确而自动地预测感知的图像质量。因为算法提供的质量的数值测度是没用的，除非它们与人类的主观性具有很好的相关性，所以在这里我们使用“预测”一次。换言之，算法应该预测出人类观察者将要报告的一幅图像的平均质量。

显然，这样的客观图像质量测度的成功制订在广阔的应用环境中具有很大的潜力。首先，它们可被用在质量控制系统中监视图像质量。例如，图像采集系统可以用一个质量指标去监视并自动调整自已来获得最好质量的图像数据。网络视频服务器可以检查网络上传输的数字视频质量来控制和分配流资源。鉴于互联网视频源最近的巨幅增长，这项应用是相当重要的。

第二，它们可被用于检测图像处理系统和算法。例如，如果可以用一些图像去噪和复原算法来增强数码相机拍摄的图像质量，那么就可以用一个质量指标来判定哪一个提供了最好的质量结果。

第三，它们可被嵌入到图像处理和传输系统中来优化系统和参数设置。例如，在视觉通信系统中，图像质量测度能在编码器的预滤波和比特分配算法的最优设计，以及在解码器的重建、错误隐藏、和后滤波算法的最优设计中给予协助。

在图像质量平均方法的设计和选择中，常常会有精度和复杂度之间的平衡问题，取决于应用场景。

例如，如果有一个客观系统能完全仿真人类视觉系统的所有相关方面，包括其内置的环境知识，那么它应该能够提供图像质量的精确预测。然而，我们有关 HVS 的知识

和我们有关环境的模型的精妙水平有限。随着我们不断提高在这些领域的知识水平，那么可以预期的是，与人类视觉机制非常接近的图像质量评价系统就会制订出来。

然而，这样的系统对人类主观质量评价的预测性能通常是很差的。实际上，尽管这些质量评价方法用作理论算法设计的解析指标是相当有用的，但长期以来对于评价真实图像的质量人们认为是很弱的，不管这幅图像是否被处理过。

C.2　通用的图像质量指标

通用图像质量指标（UIQI，Universal Image Quality Index）容易计算且可用于各种图像处理应用中。它是一个数学定义的测度，因为两个原因它很受欢迎。首先，它们容易计算且通常具有低的复杂度。其次，它们与观察条件和单个观察者无关。尽管人们认为在人类感知图像质量时观察条件起到重要的作用，但在多数情况下它们是不固定的，而且图像分析系统通常得不到特定的数据。如果有 N 种不同的观察条件，那么一个依赖观察条件的方法就会产生 N 种不同的度量结果，这是很不方便使用的。此外，衡量观察条件、计算条件参数并输入给度量系统就变成了观察者的责任了。相反，一个独立于观察条件的测度传送一个质量数值就给出了这幅图像质量如何的一般信息。

图像质量指标的通用性意味着，该方法不依赖于测试图像、观察条件或个别观察者。更重要的是，它必须可用于各种图像处理且对于不同种类的图像失真能给出有意义的对比结果。UIQI 试图取代目前广泛使用的 PSNR 和 MSE 技术。

C.2.1　质量指标的定义

令 $x = \{x_i \mid i = 1, 2, ..., N\}$ 和 $y = \{y_i \mid i = 1, 2, ..., N\}$ 分别是原始和测试图像信号。提出的质量指标定义为

$$Q = \frac{4\sigma_{xy}\overline{xy}}{(\sigma_x^2 + \sigma_y^2)\left[(\bar{x})^2 + (\bar{y})^2\right]} \tag{C.4}$$

其中，

$$\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i, \quad \bar{y} = \frac{1}{N}\sum_{i=1}^{N} y_i \tag{C.5}$$

$$\sigma_x^2 = \frac{1}{N-1}\sum_{i=1}^{N}(x_i - \bar{x})^2, \sigma_y^2 = \frac{1}{N-1}\sum_{i=1}^{N}(y_i - \bar{y})^2 \tag{C.6}$$

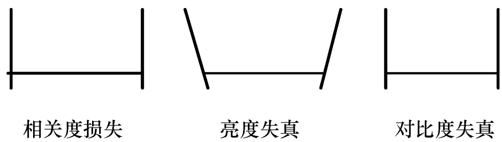
$$\sigma_{xy}^2 = \frac{1}{N-1}\sum_{i=1}^{N}(x_i - \bar{x})(y_i - \bar{y}) \tag{C.7}$$

这里 $\bar{x}$ 是原始图像的均值，$\bar{y}$ 是测试图像的均值。σ_x^2 和 σ_y^2 分别是原始和测试图像的方差。

协方差由 σ_{xy} 表示。Q 的动态范围是 $[-1, 1]$，仅当原始图像对所有的 N 值都等

于测试图像才会有最好的 Q 值 1。当测试图像是原始图像减去其均值两倍的时候，出现最差的值 -1。这个质量指标把任何失真模型化为三个因数的组合：相关度损失量、亮度失真，和对比度失真。为了理解这些，我们把 Q 的定义重写为三个分量的乘积

$$Q = \frac{\sigma_{xy}}{\sigma_x \sigma_y} \frac{2\bar{x}\,\bar{y}}{(\bar{x})^2 + (\bar{y})^2} \frac{2\sigma_x \sigma_y}{\sigma_x^2 + \sigma_y^2} \tag{C. 8}$$



第一个分量是原始和测试图像之间的相关系数，它是线性相关度的测度。它取值范围从 1 到 -1，当测试图像等于原始图像乘以一个正常数时等于最好值。在计算原始图像和测试图像的相关度之后的第二和第三分量中出现了相对失真。第二分量度量了原始和测试图像之间的平均亮度，它的取值范围是 [0，1]。当原始和测试图像的均值相等时这个分量取得最大值。信号的方差可被看作是对比度的估计值，所以第三个分量度量了两幅图像对比度的相似程度。它的取值范围也是 [0，1]，当且仅当方差相等是达到最大值。

C. 2. 2　在图像中的应用

一幅图像包括众多像素和信号，但实际上单一的总体质量值被认为是图像信号，它通常是非平稳的，因为图像质量常常是随空间而变化的。这就是局部计算统计特征然后再组合起来形成一个总体测度的原因。用于局部区域的度量方法是滑动窗口方法。大小为 $B \times B$ 的滑动窗口逐像素移动，首先从图像左上角开始水平移动，然后垂直地经过图像的所有行列直到右下角，分别如图 C. 2 ~ 图 C. 4 所示。

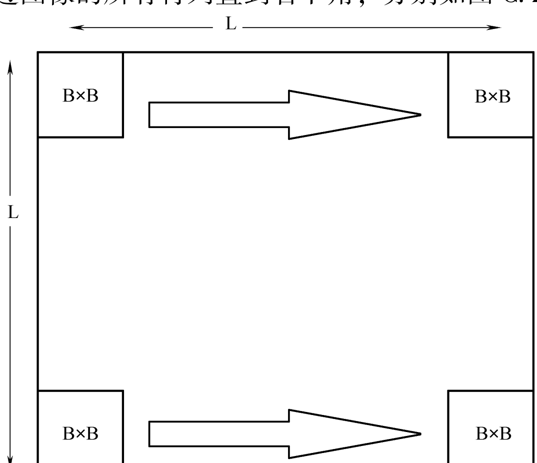


图 C. 2　滑动窗口（$B \times B$）在水平方向上位置的移动 [G11]

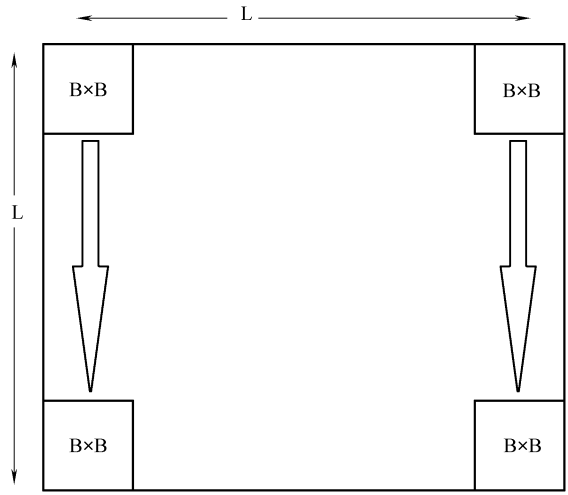


图 C.3　滑动窗口（$B\times B$）在垂直方向上位置的移动［G11］

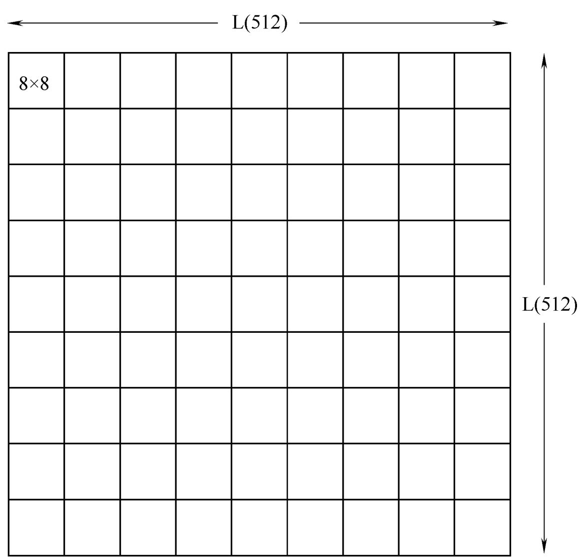


图 C.4　滑动窗口示例

滑动窗口遍历整个图像之后，就可计算总体质量指标。滑动窗口大小默认设置为 $B=8$。第 j 步的质量指标计算为 Q_j，总体质量指标如下给出

$$Q = \frac{1}{M}\sum_{j=1}^{M} Q_j$$

$$M = \text{步数} \tag{C.9}$$

对于图像大小为 $L\times L$ 滑动窗口大小为 $B\times B$ 的情况，滑动窗口遍历整个图像花费的步数 M 是$[(L-B)+1]^2$。

C.3 结构相似度指标

一个最近提出的图像保真度度量方法是 SSIM 指标，它在度量其他信号保真度方面也是高效的。作为原始 SSIM 方法基础的主要哲学是人类视觉系统［Q5］，它高度适应于从视觉场景中抽取结构信息。因此，至少为了图像保真度测量，保持信号结构应该是一项重要任务。相应地，算法可以设法度量结构失真来实现图像保真度测量。图 C.5［Q22］帮助描述结构失真和非结构失真之间的区别。在图中，非结构失真（亮度改变、对比度改变，伽玛失真和空间平移）由图像采集和显式过程中周围环境或仪器条件导致。这些失真没有改变视觉场景中对象的图像结构。然而，其他失真（加性噪声，抖动和有损压缩）明显地扭曲了对象的图像结构。如果我们把人类视觉系统看作是一个理想的信息抽取器，它寻求识别出视觉场景中的对象，那么它一定对于结构失真高度敏感而且自动补偿非结构失真。因此，一个有效的客观信号保真度测度。应该能模拟这种功能。

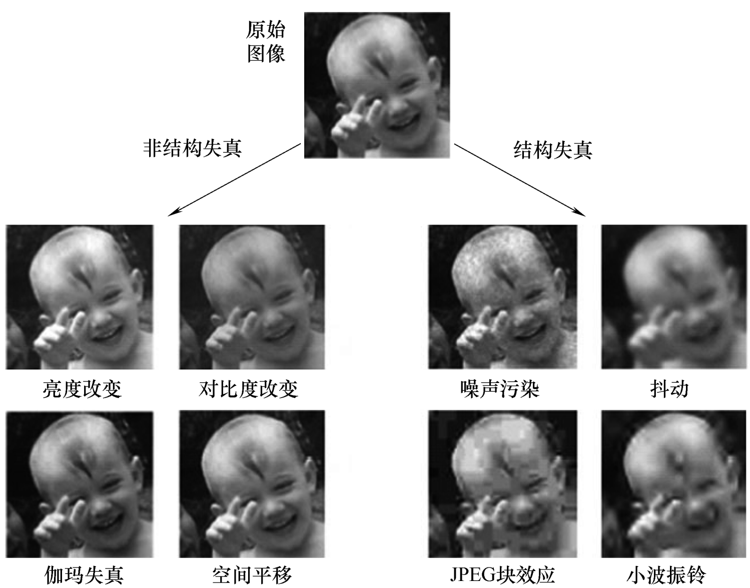


图 C.5　结构失真与非结构失真的例子［Q22］（2009　IEEE）

结构相似度测量系统框图如图 C.6 所示。假定 $x=\{x_i \mid i=1,2,...,N\}$ 和 $y=\{y_i \mid i=1,2,...,N\}$ 是两个长度有限的图像信号，而且已经彼此对齐（例如，从每幅图像中抽取的空间小块），其中 N 是信号样值（像素，如果信号是图像的话）的个数，x_i 和 y_i 分别是 $\boldsymbol{x}$ 和 $\boldsymbol{y}$ 中的第 i 个样值。

如果我们认为其中一个信号具有完美的质量，那么相似度测度就能作为第二个信号质量的定量测度。该系统把相似度度量的任务分为三个比较：亮度、对比度和结构。首先，比较每个信号的亮度。如果是离散信号，这被估计为平均强度

$$\mu_x = \frac{1}{N}\sum_{i=1}^{N} x_i \tag{C.10}$$

那么，亮度比较函数 l（$\boldsymbol{x}$，$\boldsymbol{y}$）就是μ_x和μ_y的函数。

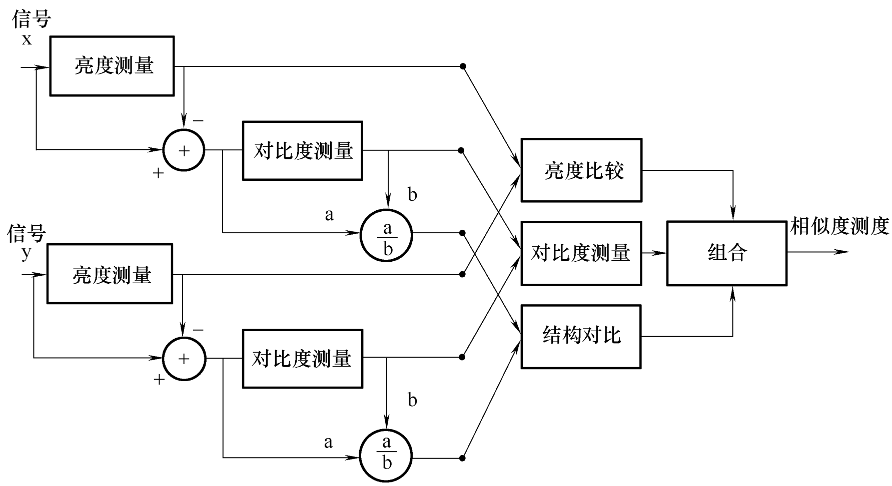


图 C.6　结构相似度测量系统框图［Q13］（2004　IEEE）

第二，我们从信号中去除平均强度。所产生离散形式的信号（$\boldsymbol{x}-\mu_x$）对应矢量在下式所定义的超平面上的投影

$$\sum_{i=1}^{N} x_i = 0 \tag{C.11}$$

式中，$\boldsymbol{x} = \{x_i \mid i=1, 2, \ldots, N\}$，是一个有限长的图像信号。

我们利用标准差（方差的平方根）作为信号对比度的估计值。离散形式的无偏估计由下式给出

$$\sigma_x = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(x_i-\mu_x)^2} \tag{C.12}$$

对比度的比较 $c(\boldsymbol{x}, \boldsymbol{y})$ 那么就是 σ_x 和 σ_y 的对比。

第三，信号由其自身的标准差进行标准化（相除），这样两个想比较的信号就具有单位标准差。在这些标准化的信号$[(\boldsymbol{x}-\mu_x)/\sigma_x]$和$[(\boldsymbol{y}-\mu_y)/\sigma_y]$上，就可以执行结构对比 $S(\boldsymbol{x}, \boldsymbol{y})$。

最后，三个分量组合起来产生一个总体相似度测度

$$S(\boldsymbol{x},\boldsymbol{y}) = f(l(\boldsymbol{x},\boldsymbol{y}), c(\boldsymbol{x},\boldsymbol{y}), s(\boldsymbol{x},\boldsymbol{y})) \tag{C.13}$$

其中

$$l(\boldsymbol{x},\boldsymbol{y})=f(\mu_x,\mu_y)$$
$$c(\boldsymbol{x},\boldsymbol{y})=f(\sigma_x,\sigma_y)$$
$$S(\boldsymbol{x},\boldsymbol{y})=f\left(\frac{\boldsymbol{x}-\mu_x}{\sigma_x}\frac{\boldsymbol{y}-\mu_y}{\sigma_y}\right)$$

重要的一点是，这三个分量是相对独立的。例如，亮度和/或对比度的改变不会影响图像的结构。为了完成（C.13）中相似度测度的定义，我们需要定义三个函数 $l(\boldsymbol{x},\boldsymbol{y})$，$c(\boldsymbol{x},\boldsymbol{y})$ 和 $s(\boldsymbol{x},\boldsymbol{y})$ 以及组合函数 $f(\cdot)$。我们还想要相似度测度满足如下特性。

1）对称性：$S(\boldsymbol{x},\boldsymbol{y})=S(\boldsymbol{y},\boldsymbol{x})$。

2）有界性：$S(\boldsymbol{x},\boldsymbol{y})\leqslant 1$。

3）最大值唯一性：$S(\boldsymbol{x},\boldsymbol{y})\leqslant 1$ 当且仅当 $\boldsymbol{x}=\boldsymbol{y}$ 时（在离散表示形式中，$x_i=y_i$ 对所有 $i=1,2,\ldots,N$）

对于亮度比较，我们定义

$$l(\boldsymbol{x},\boldsymbol{y})=\frac{2\mu_x\mu_y+C_1}{\mu_x^2+\mu_y^2+C_1} \tag{C.14}$$

其中包含的约束量 C_1 是为了避免 $\mu_x^2+\mu_y^2$ 非常接近于零时计算的不稳定。特别地，我们选择

$$C_1=(K_1L)^2 \tag{C.15}$$

式中，L 是像素值（对于 8bit 灰度级图像是 255）的动态范围，$K_1\ll 1$ 是一个小常数。类似的考虑也适用于后面描述的对比度对比和结构对比。方程（C.14）显然是遵守 SSIM 的三个特性的。

方程（C.14）也与 Weber 定律定性地一致，该定律广泛用于 HVS 中明适应（又叫作亮度掩蔽效应）的建模。按照 Weber 定律，在很宽范围的亮度值上，最小可觉（原文：just-noticeable——译者注）亮度变化 ΔI 近似正比例于背景亮度。换句话说，HVS 对相对变化敏感而不是绝对亮度变化。令 R 表示相对于背景亮度的亮度变化大小，我们把 R 重写为

$$R=\frac{\mu_y}{\mu_x}-1 \tag{C.16}$$

将 R 代入方程（C.14），给出

$$l(\boldsymbol{x},\boldsymbol{y})=\frac{2(1+R)}{1+(1+R)^2+\frac{C_1}{\mu_x^2}} \tag{C.17}$$

如果我们假定 C_1 足够小（相对于 μ_x^2）可被忽略，那么 $l(\boldsymbol{x},\boldsymbol{y})$ 就只是 R 的一个函数，与 Webe 定律定性地一致。对比度比较函数具有类似形式

$$c(\boldsymbol{x},\boldsymbol{y})=\frac{2\sigma_x\sigma_y+C_2}{\sigma_x^2\sigma_y^2+C_2} \tag{C.18}$$

式中，$C2=(K_2L)^2$，且 $K_2\ll 1$。

这一定义又一次满足三条特性。这个函数的一个重要特征是，对于同样的对比度变化量 $\Delta\sigma=\sigma_y-\sigma_x$，这一测度对于高基数对比度的情况比对于低基数对比度更不敏感。这与 HVS 的对比度掩蔽特征是一致的。

在亮度求差和方差标准化之后进行结构对比。特别地，我们将两个单位矢量 $[(\boldsymbol{x}-\mu_x)/\sigma_x]$ 和 $[(\boldsymbol{y}-\mu_y)/\sigma_y]$ 与两幅图像的结构联系起来，每个矢量都位于方程（C.11）定义的超平面上。它们之间的相关性（内积）是一种定量化结构相似度简单而有效的测度。注意，在 $[(\boldsymbol{x}-\mu_x)/\sigma_x]$ 和 $[(\boldsymbol{y}-\mu_y)/\sigma_y]$ 之间的相关度等于 $\boldsymbol{x}$ 和 $\boldsymbol{y}$ 之间的相关系数。因而，我们定义结构比较函数如下：

$$s(\boldsymbol{x},\boldsymbol{y})=\frac{\sigma_{xy}+C_3}{\sigma_x\sigma_y+C_3} \tag{C.19}$$

与对待亮度和对比度测度一样，我们在分子和分布中引入了一个小常数 C_3。σ_{xy} 的离散形式可被估计为

$$\sigma_{\mathrm{xy}}=\frac{1}{N-1}\sum_{i=1}^{N}(x_i-\mu_x)(y_i-\mu_y) \tag{C.20}$$

还应注意，$s(\boldsymbol{x},\boldsymbol{y})$ 可以渠道负值。最后，我们把三个比较结果（C.14），（C.18）和（C.19）组合起来并命名得到的信号间相似度测度为 SSIM 指标，表达如下

$$\mathrm{SSIM}(\boldsymbol{x},\boldsymbol{y})=[l(\boldsymbol{x},\boldsymbol{y})]^{\alpha}[c(\boldsymbol{x},\boldsymbol{y})]^{\beta}[s(\boldsymbol{x},\boldsymbol{y})]^{\gamma} \tag{C.21}$$

式中，$\alpha>1$，$\beta>1$，且 $\gamma>1$ 都是用来调节三个分量相对重要性的参数。容易证明，这个定义满足上面给出的三个条件。为了简化表达式，我们设置 $\alpha=\beta=\gamma=1$ 和 $C_3=C_2/2$。这样就得到了 SSIM 指标的一种特定形式

$$\begin{aligned}\mathrm{SSIM}(\boldsymbol{x},\boldsymbol{y})&=\left(\frac{2\mu_x\mu_y+C_1}{\mu_x^2+\mu_y^2+C_1}\right)\left(\frac{2\sigma_x\sigma_y+C_2}{\sigma_x^2+\sigma_y^2+C_2}\right)\left(\frac{\sigma_{xy}+C_2/2}{\sigma_x\sigma_y+C_2/2}\right)\\&=\left(\frac{2\mu_x\mu_y+C_1}{\mu_x^2+\mu_y^2+C_1}\right)\left(\frac{2\sigma_{xy}+C_2}{\sigma_x^2+\sigma_y^2+C_2}\right)\end{aligned} \tag{C.22}$$

实际上，我们常常需要整幅图像的单一总体质量测度。我们使用平均 SSIM（MSSIM）指标来评价总体图像质量：

$$\mathrm{MSSIM}(\boldsymbol{X},\boldsymbol{Y})=\frac{1}{M}\sum_{j=1}^{M}\mathrm{SSIM}(\boldsymbol{x}_j,\boldsymbol{y}_j) \tag{C.23}$$

式中，$\boldsymbol{X}$ 和 $\boldsymbol{Y}$ 分别是参考图像和失真图像；$\boldsymbol{x}_j$ 和 $\boldsymbol{y}_j$ 是在第 j 个局部窗口上的图像内容；M 是图像局部窗口的数量。

为了用 SSIM 指标度量图像质量，局部（对图像块）使用比全局（对整幅图像）使用更可取。局部窗口内计算 SSIM 是相当平凡的，窗口逐像素移动遍历整幅图像。这样的滑动窗口方法如图 C.7 所示。

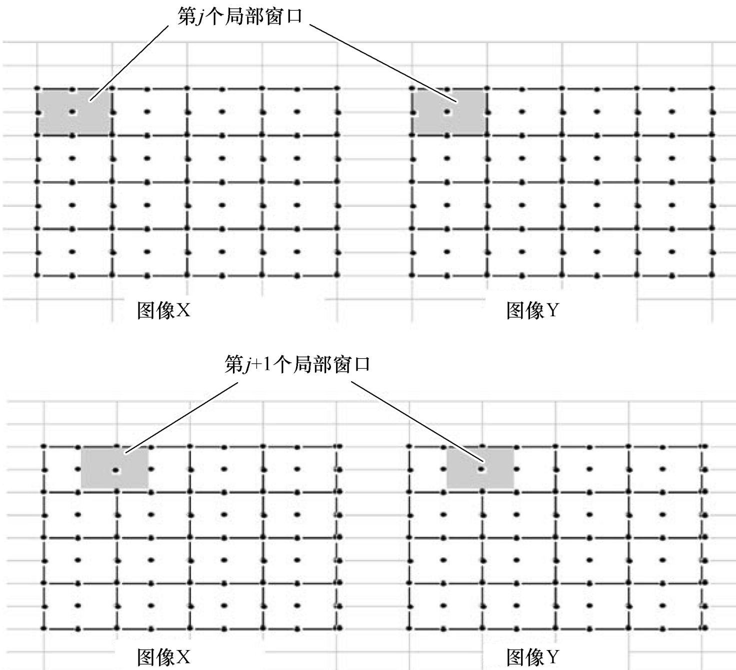


图 C.7　计算 SSIM 的滑动窗口。这里举例的滑动窗口大小为 3×3［H38］（2010　IEEE）

C.4　带有失真的图像

在本节，我们对“Lena. gif”图像加入不同类型的失真，并计算各种图像对的均方误差（MSE）和质量指标（Q）。传统的误差度量方法主要是 MSE 和峰值信噪比（PSNR，Peak Signal to Noise Ratio）。这些指标因其计算简单而且独立于观察条件和个别观察者而受到广泛使用。在另一方面，质量指标的设计是通过把任何失真模型化为三个因素的组合：相关度损失、亮度失真和对比度失真。其性能明显好于广为使用的失真指标均方误差（见图 C.8）。

1）椒盐噪声：其本身表现为随机出现的白色和黑色像素。对于这种类型噪声的一种有效的去噪方法是使用中值滤波器。在快速瞬变现象如错误切换发生的情况下，椒盐噪声就会出现在图像中。因为椒盐噪声而失真的图像看起来就像图 C.9 那样。

2）乘性斑点噪声：斑点噪声是一种颗粒噪声，固有存在于图像中并降低了图像质量。斑点噪声是一种乘性噪声，即它在任何区域都是正比于局部灰度级的。信号和噪声彼此统计独立。信号元素的均值和方差等于局部区域的均值和方差，而区域的中心就是那个像素（见图 C.10）。



图 C. 8　Lena 原始图像



图 C. 9　椒盐噪声［Q29］



图 C. 10　乘性斑点噪声［Q29］

3）图像抖动：图像的抖动通常使得图像不能聚焦。在图像处理中，一般可以通过令图像和一个低通滤波器进行卷积来获得。在这个特例中，我们用高斯抖动（见图 C. 11）。

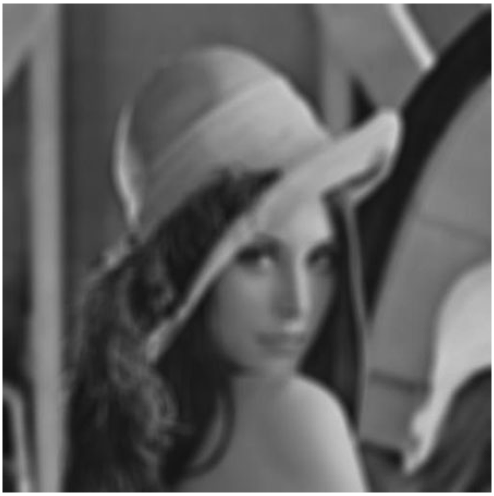


图 C. 11 图像抖动［Q29］

4）对比度拉伸：出现低对比度图像常常是因为光照条件弱或非均匀，或成像传感器的动态范围小。它可被表达为

$$V=\alpha u \qquad 0\leqslant u<a \tag{C. 24a}$$

$$=\beta(u--a)+\mathrm{va}, \quad a\leqslant u<b \tag{C. 24b}$$

$$=\gamma(u--b)+\mathrm{vb}, \quad b\leqslant u<L \tag{C. 24c}$$

变换的斜率要选择得大于整个拉伸区域。参数 a 和 b 可以通过检查图像直方图来得到。例如，像素最频繁出现的灰度级范围最可能被拉伸以改善场景的整体可见度。这里，我们已把 α 考虑为大于 1（见图 C. 12）。

5）高斯噪声：高斯噪声是一种统计噪声，它具有正态分布的概率密度函数（也称为高斯分布）。换句话说，噪声的取值是高斯分布的。它经常用作加性白噪声以生成加性高斯白噪声（AWGN，Additive White Guass Noise）（见图 C. 13）

6）均值平移算法：举止平移算法通过让每个点与数据集的概率密度峰值建立联系对 n 维数据集（也就是说，每个数据点由一个 n 维矢量来描述）进行分类。对每个点，均值平移法通过定义一个半径为 r 的球形窗口并计算位于该窗口内的点的均值计算其相关的峰值。然后算法把窗口平移到均值并重复直到收敛，也就是说，直到平移低于一个阈值（例如 0. 01）。在每次迭代过程中，窗口会移动到一个更为密集分布的数据集部分直到达到一个峰值，在峰值处数据在窗口内均匀分布（见图 C. 14）。



图 C. 12　对比度拉伸［Q29］



图 C. 13　高斯噪声［Q29］



图 C. 14　均值平移算法［Q29］

7）JPEG 压缩：原始图像的大小受到压缩，图像的 MSE 被计算出来。这种情况的压缩比是 4. 8574。用 JPEG 进行压缩又要保持一个特定 MSE 是很困难的。尽管仔细观察这些图像可能会发现三种视觉问题：块效应、抖动和振铃，但 JPEG 的第一维度解决方案更为主要。为了进行 JPEG 压缩原始图像要转换成单色度图像，然后使用 JPEG 编译码（独立软件组［JP9］）公用软件包对图像进行编码（见图 C. 15 和表 C. 1）。

C. 5 测试结果

失真的 Lena 图像：见表 C. 1 和图 C. 16；
失真的 Goldhill 图像：见表 C. 2 和图 C. 17；
失真的 Couple 图像：见表 C. 3 和图 C. 18。



图 C. 15　JPEG 压缩［Q29］

表 C. 1　Lena 图像测试结果列表［Q29］

序　号	图　　像	均 方 误 差	PSNR/dB	Q	SSIM	MSSIM
1	原始图像	0	无穷大	1	1	1
2	均值平移	224. 9993	24. 6090	0. 9894	0. 9890	0. 9894
3	对比度拉伸	225. 0932	24. 6072	0. 9372	0. 9494	0. 9470
4	脉冲椒盐噪声	225. 3684	24. 6019	0. 6494	0. 7227	0. 6723
5	乘性斑点噪声	224. 7482	24. 6138	0. 4408	0. 5009	0. 4883
6	加性高斯噪声	225. 1804	24. 6055	0. 3891	0. 4508	0. 4390
7	抖动	224. 1397	24. 6256	0. 3461	0. 6880	0. 6007
8	JPEG 压缩	215. 1139	24. 8041	0. 2876	0. 6709	0. 5572

表 C.2　图像 Goldhill 的测试结果列表［Q29］

序　号	图　　像	均 方 误 差	PSNR/dB	Q	SSIM	MSSIM
1	原始图像	0	无穷大	1	1	1
2	均值平移	121	27.3029	0.9928	0.9927	0.9929
3	对比度拉伸	120.9002	27.3065	0.9498	0.9698	0.9672
4	脉冲椒盐噪声	120.2122	27.3313	0.8290	0.8643	0.8402
5	乘性斑点噪声	121.4297	27.2876	0.6758	0.7032	0.7067
6	加性高斯噪声	121.1260	27.2984	0.6151	0.6556	0.6553
7	抖动	121.9371	27.2694	0.5080	0.6671	0.6372
8	JPEG 压缩	117.4739	27.4314	0.4963	0.6824	0.6385



原始图像,MSE=0,
Q=1,PSNR-inf dB
SSIM=1,MSSIM=1



均值平移,MSE=224.9933,
Q=0.9894,PSNR=24.6090dB
SSIM=0.9890,MSSIM=0.9894



对比度拉伸,MSSIM=0.9470,
MSE=225.0932,Q=0.9372,
PSNR=24.6072dB SSIM=0.9494



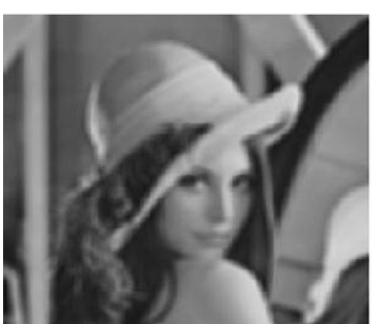
脉冲椒盐噪声,MSE=225.0932,
Q=0.6494,PSNR=24.6019dB
SSIM=0.7227,MSSIM=0.6723



乘性斑点噪声,MSE=224.7482,
Q=0.448,PSNR=24.61dB
SSIM=0.5009,MSSIM=0.4883



加性高斯噪声,MSE=225.1804,
Q=0.3891,PSNR=24.6055dB
SSIM=0.4508,MSSIM=0.4390



抖动,MSE=224.1397,
Q=0.3461,PSNR=24.6256dB
SSIM=0.6880,MSSIM=0.6



JPEG压缩,MSE=215.1139
Q=0.2876,PSNR=24.8041dB
SSIM=0.6709,MSSIM=0.5572

图 C.16　有失真的 Lena 图像

表 C.3　Couple 图像测试结果列表［Q29］

序　号	图　　像	均 方 误 差	PSNR/dB	Q	SSIM	MSSIM
1	原始图像	0	无穷大	1	1	1
2	均值平移	80.9952	29.0462	0.9947	0.9940	0.9947
3	对比度拉伸	80.9063	29.0510	0.9621	0.9631	0.9657
4	脉冲椒盐噪声	80.8358	29.0548	0.8806	0.9057	0.8874
5	乘性斑点噪声	81.2837	29.0308	0.7183	0.7613	0.7553
6	加性高斯噪声	80.6841	29.0629	0.7039	0.7511	0.7444
7	抖动	81.2747	29.0313	0.7500	0.8329	0.8238
8	JPEG 压缩	81.9302	28.994	0.6761	0.8013	0.7771



图 C.17　有失真的 Goldhill 图像

C.6　结论

Q 是简单的，而且与传统的 MSE 和 PSNR 相比，数学模型似乎是一个更好的图像质量测度。这种成功是由于 Q 度量结构失真的强大能力，结构失真发生在图像退化过程中，与 MSE 相比它对误差能量更为敏感。毫无疑问，HVS 的精确建模总是好一些，但良好定义的模型，其数学框架会逐步完善以实现成功的质量指标。

从图像形成的角度来看，SSIM 指标是结构相似度哲学的一种特定实现。SSIM



原始图像,MSE=0,PSNR=INF dB,Q=1,SSIM=1,MSSIM=1



均值平移,MSE=80.9952,PSNR=29.0462dB,Q=0.9947,SSIM=0.9940,MSSIM=0.9947



对比度拉伸,MSE=80.9063,PSNR=29.0510dB,Q=0.9621,SSIM=0.9631 MSSIM=0.9657



脉冲椒盐噪声,MSE=80.8358,PSNR=29.0548dB,Q=0.8806,SSIM=0.9057



乘性斑点噪声,MSE=81.2837,PSNR=29.0308dB,Q=0.7183,SSIM=0.7613 MSSIM=0.7553



加性高斯噪声,MSE=80.6841,PSNR=29.0629dB,Q=0.7039,SSIM=0.7511 MSSIM=0.7444



抖动,MSE=81.2747,PSNR=29.0313dB,Q=0.7500,SSIM=0.8329 MSSIM=0.8238



JPEG压缩,MSE=81.9302,PSNR=28.994dB,Q=0.6761,SSIM=0.8013 MSSIM=0.7771

图 C.18　带有失真的 Couple 图像

的关键成功是结构信息和结构失真的概念。

SSIM 指标呈现了度量失真时的输入相关行为。在某些失真情况下如抖动和 JPEG 压缩，MSSIM 是一个比 UIQI 更好的指标，这是由于 SSIM 中包括了避免不稳定性的常数 C_1 和 C_2。即使不同失真的 MSE 是相同的，SSIM 和 MSSIM 真正地代表了视觉（感知）质量。读者可以参考［Q3，Q6，Q7，Q9，Q12，Q20，Q34，U4］。

专题研究 C　图像知觉质量评价指标：结构相似度

本专题关于结构相似度指标，它表示基于结构信息的知觉图像质量。SSIM 是一个客观图像质量指标且优于传统的定量测度诸如 MSE 和 PSNR。本专题说明了基于 SSIM 的图像质量评价并描述了针对人类视觉知觉的有效性。通过回顾下面列出的论文，将非常有助于理解 SSIM 及其应用。

1）SSIM 的一般形式是

$$\mathrm{SSIM}(\boldsymbol{x},\boldsymbol{y}) = [l(\boldsymbol{x},\boldsymbol{y})]^{\alpha}[c(\boldsymbol{x},\boldsymbol{y})]^{\beta}[S(\boldsymbol{x},\boldsymbol{y})]^{\gamma}$$

注意，$\alpha>0$、$\beta>0$ 和 $\gamma>0$ 都是用来调节三个分量相对重要性的参数。其中 $\boldsymbol{x}$，$\boldsymbol{y}$ 是图像区块，$l(\boldsymbol{x},\boldsymbol{y})=\dfrac{2\mu_x\mu_y+C_1}{\mu_x^2+\mu_y^2+C_1}$，$c(\boldsymbol{x},\boldsymbol{y})=\dfrac{2\sigma_x\sigma_y+C_2}{\sigma_x^2+\sigma_y^2+C_1}$，$S(\boldsymbol{x},\boldsymbol{y})=\dfrac{\sigma_{xy}+C_3}{\sigma_x\sigma_y+C_3}$，$l(\boldsymbol{x},\boldsymbol{y})$ 是亮度比较结果、$c(\boldsymbol{x},\boldsymbol{y})$ 是对比度比较结果，$S(\boldsymbol{x},\boldsymbol{y})$ 是结构对比结果。C_1，C_2，C_3 是常数。μ_x，μ_y，σ_x，σ_y，σ_{xy} 如参考文献［Q13］中方程 14，15，16 所定义。高斯

加权函数具有形式 $w(n_1, n_2) = \exp\left(ndash; \frac{n_1^2 + n_2^2}{2\sigma^2}\right), n_1, n_2 = 1, 2, \cdots, 11$。

请写一个 Matlab 函数 my_ssim，其输入是两幅图像和 α，β，γ，C_1，C_2，C_3，输出是两幅图像的 SSIM_metric 和 SSIM_map。SSIM_metric 是 SSIM_map 的均值，它由一个局部滑动窗口（11 × 11 像素）来计算。您可以从 https://ece. uwaterloo. ca/ ~ z70wang/research/ssim 下载 SSIM 的 Matlab 实现程序作为参考。

2）像论文中那样，设置 α，β，γ，C_1，C_2，C_3，并将您的函数用于不同失真但均方误差（MSE）相同的 Lena 图像（512 × 512）。测试图像可以在链接 https://ece. uwaterloo. ca/ ~ z70wang 中 "Universal image quality index" 处下载。请计算 SSIM_metric 并给出 SSIM_map。您会发现与 MSE 相比，SSIM_metric 与人类的质量知觉更为相关。

3）与上一条一样将 C_1，C_2，C_3 固定并设 α 为 1。任选 5 对 β，γ，例如 [$\beta = 1$，$\gamma = 2$]，[$\beta = 1$，$\gamma = 3$]，[$\beta = 1$，$\gamma = 4$]，[$\beta = 1$，$\gamma = 4$]，[$\beta = 3$，$\gamma = 1$]，并将您的函数作用于失真的 Lena 图像（您可用所有或部分图像）。然后找出产生与您对图像质量感知最相关的 SSIM_metric。没有争取答案，完全取决于您自己对图像质量的意见。

2008 年 4 月，JVT 汉诺威会议引入了 SSIM [Q20]。本附录包括了整个 ITU-T 文档。

附录 D　H.264 中模式相关的 DCT/DST 实现方案

本附录基于 Priyadarshini Anjanappa［E111］的学位论文。

欲知详情，请参见 UTA 的 EE5359 课程网址 http://www-ee.uta.edu/Dip/Courses/EE5359/index.html 上的 Thesis/Project 标题。

D.1　前言

模式相关的 DCT/DST 是一种混合变换编码方案，将正弦和余弦变换之间的切换纳入到了帧内预测模式，从而开发块间的相关性。从一侧预测时，预测残差能量随着我们远离边界而增加。正弦变换较好地适应于这种预测残差统计规律［E130］。已经证明，帧内预测后，沿着垂直（水平）预测方向的垂直（水平）模式的最优变换就是［E129］［E120］中的自相关矩阵的KLT，用如下叫作 DST-VII 的正弦变换来计算：

$$[T_S]_{i,j} = \frac{2}{\sqrt{2N+1}} \sin \frac{(2i-1)j\pi}{2N+1}$$

不断发展的 HEVC 标准化过程使用统一的帧内预测，其中多达 34 种不同方向的帧内预测模式可被分为如下 3 类：

- 第 1 类斜向模式如图 D.1 所示：这里或者从上方的行或者从左侧的列中的解码像素中进行预测。垂直模式和水平模式是这种斜向模式的特例，分别发生在预测方向是垂直或水平的时候；
- 第 2 类斜向模式如图 D.1 所示：这里，既从上方行又从左侧列的像素来进行预测；
- 直流模式：这里，从所有得到的解码像素平均值中进行预测。

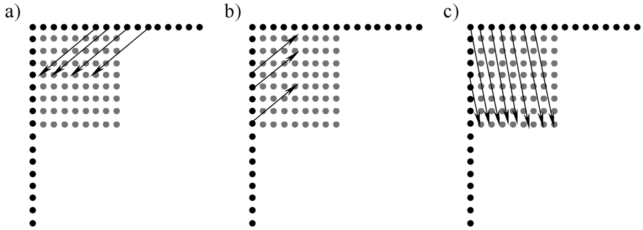


图 D.1　a）第 1 类斜向模式：只从上方行预测
b）第 1 类斜向模式：只从左侧列预测 c）第 2 类斜向模式［E131］

从统一帧内预测模式到用于 HEVC 标准化［E130］［E131］早期阶段的 DCT/DST 的映射如表 D.1 所示。

表 D.1 从帧内预测模式到用于 HM2.0 的 DCT/DST 的映射

模　式	统一帧内预测方向	垂　直	水平变换
0,5,6,12,13,22,23,24,25	VER to VER+8	DST	DCT
1,8,9,16,17,30,31,32,33	HOR to HOR+8	DCT	DCT
2	DC	DCT	DST
3,4,10,11,18,19,20,21	VER−8 to VER−1	DST	DCT
7,14,15,26,27,28,29	HOR−8 to HOR−1	DST	DST

D.2 参考软件中变换的实现方案

JM 软件［H30］是 H.264/AVC 标准的参考软件。图 D.2 给出了 DCT 正变换和逆变换的快速实现方案的流程图，在 JM 软件［H4］中它用于每个 4×4 块的行和列。不需要乘法，只要加法和移位运算。用在 H.264 的 DCT 是 DCT-II 的整数近似，通过一个线性变换 $\boldsymbol{X}=\boldsymbol{H}\boldsymbol{x}$ 把一个 N 长的矢量 $\boldsymbol{x}$ 映射到一个新的变换系数矢量 $\boldsymbol{X}$，其中 $\boldsymbol{H}$ 第 k 行第 n 列的元素定义如下

$$\boldsymbol{H}(k,n)=c_{\mathrm{k}}\sqrt{\frac{2}{N}}\cos\left[\left(n+\frac{1}{2}\right)\frac{k\pi}{N}\right]$$

式中，$k=0,1,\cdots,N-1$，是频率下标；$n=0,1,\cdots,N-1$ 是样值下标，对 $k>0$，$c_0=\sqrt{2}$和 $c_{\mathrm{k}}=1$。

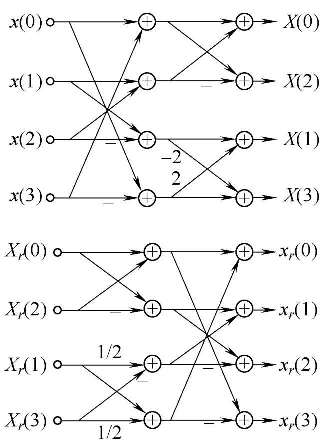


图 D.2 H.264 的 DCT 正变换（上）和逆变换（下）的快速实现［H4］

HM 软件［E54］是 HEVC 的参考软件。用于 HM 软件的 DST-VII 快速实现包

括加法、乘法和移位运算。

D.3　提出的方案

D.3.1　从帧内预测模式到 DCT/DST 的映射

类似于 HEVC 的映射，H.264 的 9 种面向 4×4 亮度块的帧内预测模式可被分类为第一类斜向模式，第二类斜向模式和直流模式。提出的 H.264/AVC 帧内预测模式的映射见表 D.2。

D.3.2　获得 H.264 的 DST 矩阵

H.264/AVC 的 DCT 正变换矩阵是

$$\text{正变换 DCT} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{bmatrix}$$

表 D.2　提出的从帧内预测模式到 H.264/AVC 中 DCT/DST 的映射方案

模　　式	帧内预测方向	垂直变换	水平变换
0，3，7	垂直，对角左下，垂直向右	DST	DCT
1，8	水平，水平向上	DCT	DST
2	直流	DCT	DCT
4，5，6	对角右下，垂直向左，水平向下	DST	DST

DCT 正变换矩阵行基矢量范数：

第 1 行：2

第 2 行：$\sqrt{10}=3.1623$

第 3 行：2

第 4 行：$\sqrt{10}=3.1623$

用在 H.264/AVC 中的 DCT 变换逆矩阵是

$$\text{逆变换 DCT} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1/2 & -1/2 & -1 \\ 1 & -1 & -1 & 1 \\ 1/2 & -1 & 1 & -1/2 \end{bmatrix}$$

DCT 逆变换矩阵行基矢量范数：

第 1 行：2

第 2 行：$\sqrt{5/2}=1.5811$

第 3 行：2

第 4 行：$\sqrt{5/2}=1.5811$

用于 HEVC 的 DST 矩阵是

$$\text{DST 矩阵}=\begin{bmatrix}29 & 55 & 74 & 84\\ 74 & 74 & 0 & -74\\ 84 & -29 & -74 & 55\\ 55 & -84 & 74 & -29\end{bmatrix}$$

用于 HEVC 的 DST 矩阵行基矢量范数：

第 1 行：$\sqrt{16398}=128.0547$

第 2 行：$\sqrt{16498}=128.1718$

第 3 行：$\sqrt{16398}=128.0547$

第 4 行：$\sqrt{16398}=128.0547$

为了在 H.264 中执行同样的 DST，要使沿着 DST 矩阵每一行的范数等于 DCT 矩阵相应行的范数。

为了得到 H.264 的 DST 正变换矩阵，要让 HEVC 所用的 DST 的行除以因数

$$\frac{\text{DST 矩阵行“}x\text{”的范数}}{\text{DCT 正变换矩阵行“}x\text{”的范数}}$$

H.264 所获得的改良 DST 正变换矩阵是

$$\text{正变换 DST}=\begin{bmatrix}0.4529 & 0.8590 & 1.1558 & 1.3119\\ 1.8257 & 1.8257 & 0 & -1.8257\\ 1.3119 & -0.4529 & -1.1558 & 0.8590\\ 1.3582 & -2.0744 & 1.8274 & -0.7161\end{bmatrix}$$

改良 DST 正变换矩阵的行基矢量的范数：

第 1 行：2

第 2 行：3.1622

第 3 行：2

第 4 行：3.1622

可以看出，改良 DST 正变换矩阵行矢量范数与 DCT 正变换矩阵相同。

为了得出 H.264 的 DST 逆变换矩阵，将 HEVC 所用的 DST 矩阵行除以因数

$$\frac{\text{DST 矩阵行“}x\text{”的范数}}{\text{DCT 逆变换矩阵行“}x\text{”的范数}}$$

H.264 所用的改良 DST 逆变换矩阵是：

$$
\text{逆变换 DST} = \begin{bmatrix} 0.4529 & 0.8590 & 1.1558 & 1.3119 \\ 0.9129 & 0.9129 & 0 & -0.9129 \\ 0.4529 & -0.8590 & -1.1558 & 1.3119 \\ 0.6791 & -1.0372 & 0.9137 & -0.3581 \end{bmatrix}
$$

改良 DST 逆变换矩阵各行的基矢量范数：

第 1 行：2

第 2 行：1.5812

第 3 行：2

第 4 行：1.5812

可以看到，改良 DST 正变换矩阵的各行范数与 DCT 正变换矩阵相同。

D.3.3 H.264/AVC 参考软件中 DCT/DST 的实现

改良 DST 矩阵不是整数矩阵，因而 H.264 参考软件 JM 18.4 [H30] 执行的是矩阵乘法而不是快速实现算法。已经证明，矩阵乘法能给出像 H.264 整数 DCT 快速实现一样精确的结果。主档次用于视频序列。编码视频序列可以从 [V5] 下载。编码过程用处理器为 Intel i7 Quad 4，2.0 GHz，内存为 8GB 的系统来执行。所用的操作系统是 Windows 7。HEVC 默认快速实现方案的性能分析也使用 HM8.0，[E58] 视频序列用帧内主档次进行编码。用于评价性能的视频序列有 HD（1920×1080，1280×720），WVGA（832×480）和 WQVGA（416×240）[G16]。只有视频序列的第一帧（I 帧）进行编码。每个序列的编码结果——比特率对 PSNR，比特率节约百分比（BD-比特率）和绝对 PSNR 增益（BD-PSNR）——都列于表中。对于每个序列还给出了不同分辨率下的 RD 曲线图。BD-PSNR 和 BD-比特率 [E79] [Q32] 用于分析 RD（率失真）曲线图的性能。负值的 BD-比特率或者正值的 BD-PSNR 表示较好的编码效率。

为了分析的需要，JM18.4 [H30] 编码器配置文件改变了下列参数设置：

- FramesToBeEncoded = 1
- SourceWidth = 416，832，1280，1920
- SourceHeight = 240，480，720，1080
- OutputWidth = 416，832，1280，1920
- OutputHeight = 240，480，720，1080
- FrameRate = 30
- ProfileIDC = 77
- LevelIDC = 50
- IntraPeriod = 0
- IDRPeriod = 0
- QPISlice = 16，20，24，28

- RDOptimization = 1
- Transform8x8Mode = 0.

为了分析的需要，HM8.0［E54］编码器配置文件改变了下列参数设置：

- IntraPeriod = −1
- QP = 16，20，24，28
- RDOQ = 1

为了分析的需要，HM8.0［E54］输入序列配置文件改变了下列参数设置：

- FrameRate = 30
- FramesToBeEncoded = 1
- SourceWidth = 416，832，1280，1920
- SourceHeight = 240，480，720，1080

D.4 BD-PSNR 和 BD-比特率的计算

BD-PSNR 和 BD-比特率分别表示两幅 RD 曲线图之间的平均 PSNR 和比特率之差［E79］。

- 通过对应于 QP = 16，20，24，28 的四个数据点拟合一个曲线图
- 数据点是 BD PSNR 的 Y-PSNR 值和 BD 率的比特率值
- 通过这四个数据点由形如下式的三次多项式可得一个很好的内插曲线

$$\text{SNR} = a + b \times \text{bit} + c \times \text{bit}^2 + d \times \text{bit}^3$$

其中要确定 a，b，c，d 使得曲线通过所有的数据点

- RD 曲线之间的插值由高比特率主导。因此考虑比特率的对数值来计算 BD 率
- 同样，插值可以作为 SNR 的函数来完成如下：

$$\text{bit} = a + b \times \text{SNR} + c \times \text{SNR}^2 + d \times \text{SNR}^3$$

- 基于内插，找到一个曲线积分表达式
- 平均差值是积分之差除以积分间隔

来自［Q32］的计算 BD-PSNR 和 BD-比特率的 MATLAB 实现程序用于以上分析。

D.5 性能分析

D.5.1 WQVGA（416×240）序列的测试结果

WQVGA（416×240）序列的测试结果分别见表 D.3 ~ 表 D.6 和图 D.3 ~ 图 D.5。

表 D. 3　三个 416 × 240 序列比特率和 PSNR 的比较

（用 DCT/DST 的 H. 264/AVC 与默认 H. 264/AVC 相比）

序列名称	QP	默认 H. 264/AVC 比特率 Y-PSNRU-PSNRV-PSNR (kbit/s)(dB)	用 DCT/DST 的 H. 264/AVC 比特率 Y-PSNRU-PSNRV-PSNR (kbit/s)(dB)
RaceHorses	16	10105. 6847. 74447. 20147. 377	10306. 3247. 15547. 20147. 377
	20	7556. 1644. 08444. 03044. 497	7662. 7243. 61744. 03044. 497
	24	5429. 7640. 49641. 31341. 726	5476. 0840. 19841. 31341. 726
	28	3792. 2437. 09738. 89439. 324	3801. 1236. 93638. 89439. 324
BlowingBubbles	16	11159. 5247. 20246. 29247. 172	11401. 246. 49446. 29247. 172
	20	8124. 0043. 12743. 18744. 521	8249. 0442. 66743. 18744. 521
	24	5575. 9239. 53640. 61541. 987	5617. 6839. 25340. 61541. 987
	28	3744. 0036. 38538. 32339. 879	3730. 8036. 23438. 32339. 879
BQSquare	16	11502. 7247. 33747. 54047. 689	11832. 4846. 70447. 54147. 689
	20	8756. 6443. 55044. 76145. 304	8965. 4443. 09044. 76145. 304
	24	6485. 7639. 94842. 49943. 097	6625. 2039. 48342. 49943. 097
	28	4740. 7236. 51740. 76141. 378	4814. 1636. 22740. 76141. 378

表 D. 4　BD-PSNR 与 BD-比特率

（用 DCT/DST 的 H. 264/AVC 与默认 H. 264/AVC 相比）

序列名称	BD-PSNR/dB	BD-比特率(%)
RaceHorses	−0. 4913	4. 775
BlowingBubbles	−0. 4852	5. 1894
BQSquare	−0. 7315	6. 2691

表 D. 5　三个 416 × 240 序列的比特率和 PSNR 的比较

（H. 264/AVC 与 HEVC）

序列名称	QP	默认 H. 264/AVC 比特率 Y-PSNRU-PSNRV-PSNR (kbit/s)(dB)	默认 HEVC 比特率 Y-PSNRU-PSNRV-PSNR (kbit/s)(dB)
RaceHorses	16	10105. 6847. 74447. 20147. 377	8411. 5247. 301747. 243647. 3423
	20	7556. 1644. 08444. 03044. 497	6175. 2043. 826444. 370044. 7900
	24	5429. 7640. 49641. 31341. 726	4414. 5640. 540341. 552642. 2303
	28	3792. 2437. 09738. 89439. 324	2962. 8037. 159239. 077839. 5112
BlowingBubbles	16	11159. 5247. 20246. 29247. 172	9757. 9246. 870746. 163647. 3226
	20	8124. 0043. 12743. 18744. 521	6872. 8842. 840743. 421644. 8203
	24	5575. 9239. 53640. 61541. 987	4662. 9639. 453840. 893842. 4269
	28	3744. 0036. 38538. 32339. 879	2977. 4436. 322038. 330139. 8608
BQSquare	16	11502. 7247. 33747. 54047. 689	9799. 9246. 900347. 691247. 8703
	20	8756. 6443. 55044. 76145. 304	7368. 4843. 245944. 845345. 4624
	24	6485. 7639. 94842. 49943. 097	5380. 5639. 723942. 449743. 1106
	28	4740. 7236. 51740. 76141. 378	3788. 8836. 268640. 492941. 1079

表 D.6 三个 416×240 序列的 BD-PSNR 和 BD-比特率

（H.264/AVC 与 HEVC）

序列名称	BD-PSNR/dB	BD-比特率(%)
RaceHorses	2.0212	-17.735
BlowingBubbles	1.4722	-14.2433
BQSquare	1.8785	-14.6427

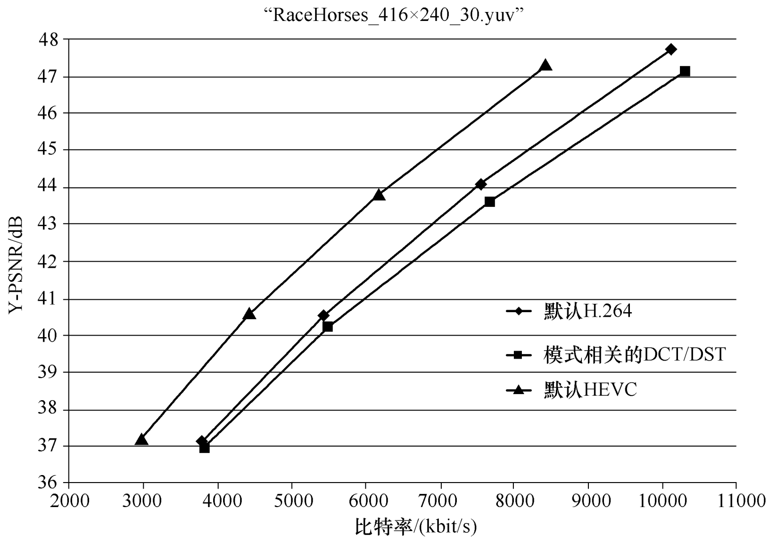


图 D.3 RaceHorses 序列 Y-PSNR 随比特率的变化

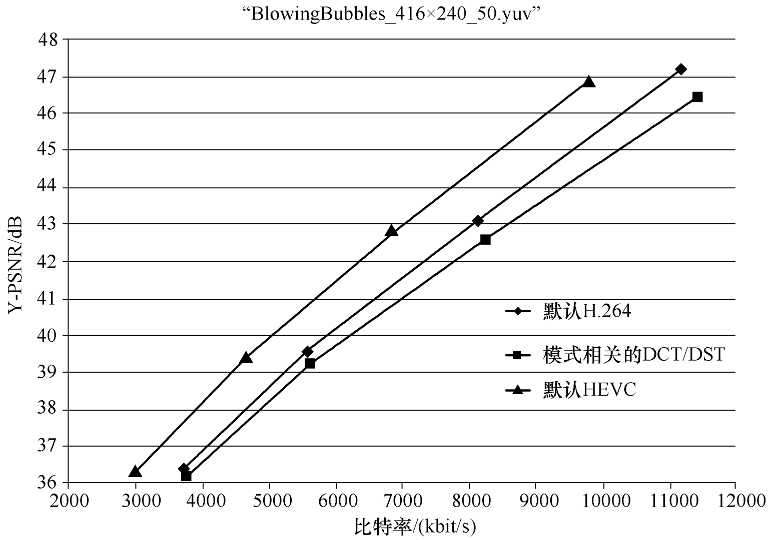


图 D.4 BlowingBubbles 序列 Y-PSNR 随比特率的变化曲线

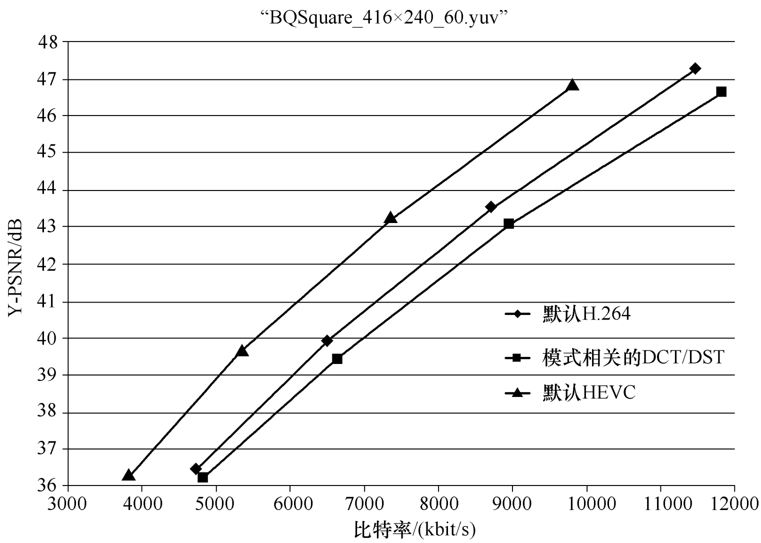


图 D. 5　BQSquare 序列 Y-PSNR 随比特率的变化曲线

D. 5. 2　WVGA（832 ×480）序列的测试结果

WVGA（832 × 480）序列的测试结果分别见表 D. 7 ~ 表 D. 10 和图 D. 6 ~ 图 D. 8。

表 D. 7　三个 832 ×480 序列的比特率和 PSNR 比较

（用 DCT/DST 的 H. 264/AVC 与默认 H. 264/AVC 相比）

序列名称	QP	默认 H. 264/AVC 比特率 Y-PSNRU-PSNRV-PSNR (kbit/s)(dB)	使用 DCT/DST 的 H. 264/AVC 比特率 Y-PSNRU-PSNRV-PSNR (kbit/s)(dB)
BQMall	16	33409. 6846. 85846. 7947. 763	34025. 2846. 34346. 79047. 763
	20	22413. 8443. 27044. 40645. 660	22548. 4842. 96744. 40645. 660
	24	14934. 9640. 54342. 35243. 492	14905. 2040. 27842. 35243. 492
	28	10274. 4037. 90140. 49641. 472	10238. 1637. 64140. 49641. 472
Keiba	16	23370. 2447. 05448. 83149. 288	22886. 8846. 62648. 83149. 288
	20	15057. 3643. 80846. 65947. 422	14381. 0443. 61146. 65947. 422
	24	9540. 2441. 21744. 52845. 296	9150. 2441. 13644. 52845. 296
	28	6162. 4838. 68342. 59443. 321	5952. 2438. 59442. 59443. 321
PartyScene	16	52926. 9647. 39746. 54747. 006	54041. 2846. 56946. 54747. 006
	20	40601. 2843. 34543. 24943. 794	41196. 0042. 70543. 24943. 794
	24	29612. 1639. 49240. 32340. 846	29968. 0839. 06040. 32340. 846
	28	20975. 7636. 07337. 93538. 350	21224. 4035. 76037. 93538. 350

表 D.8 三个 832×480 序列的 BD-PSNR 和 BD-比特率比较
(用 DCT/DST 的 H.264/AVC 与默认 H.264/AVC 相比)

序列名称	BD-PSNR/dB	BD-比特率(%)
BQMall	-0.3462	4.6475
Keiba	0.0665	-1.1473
PartyScene	-0.6996	6.0669

表 D.9 三个 832×480 序列的比特率和 PSNR 比较(H.264/AVC 与 HEVC)

序列名称	QP	默认 H.264/AVC 比特率 Y-PSNRU-PSNRV-PSNR (kbit/s)(dB)	默认 HEVC 比特率 Y-PSNRU-PSNRV-PSNR (kbit/s)(dB)
BQMall	16	33409.6846.85846.7947.763	28161.8446.516546.618447.8058
	20	22413.8443.27044.40645.660	17480.4042.923644.466845.7898
	24	14934.9640.54342.35243.492	11940.2440.467842.417543.6628
	28	10274.4037.90140.49641.472	8022.4837.887140.457541.4867
Keiba	16	23370.2447.05448.83149.288	19457.2846.768049.037649.5021
	20	15057.3643.80846.65947.422	11888.6443.723646.839147.4610
	24	9540.2441.21744.52845.296	7589.7641.327244.668445.4265
	28	6162.4838.68342.59443.321	4845.3638.965242.505243.2928
PartyScene	16	52926.9647.39746.54747.006	47571.6047.083946.243946.7408
	20	40601.2843.34543.24943.794	35791.4442.979643.382543.9579
	24	29612.1639.49240.32340.846	25742.4039.225640.608441.0488
	28	20975.7636.07337.93538.350	17608.5635.777237.993838.4236

表 D.10 三个 832×480 序列的 BD-PSNR 和 BD-比特率比较(H.264/AVC 与 HEVC)

序列名称	BD-PSNR/dB	BD-比特率(%)
BQMall	1.4754	-17.985
Keiba	1.3704	-20.3635
PartyScene	1.2773	-10.2026

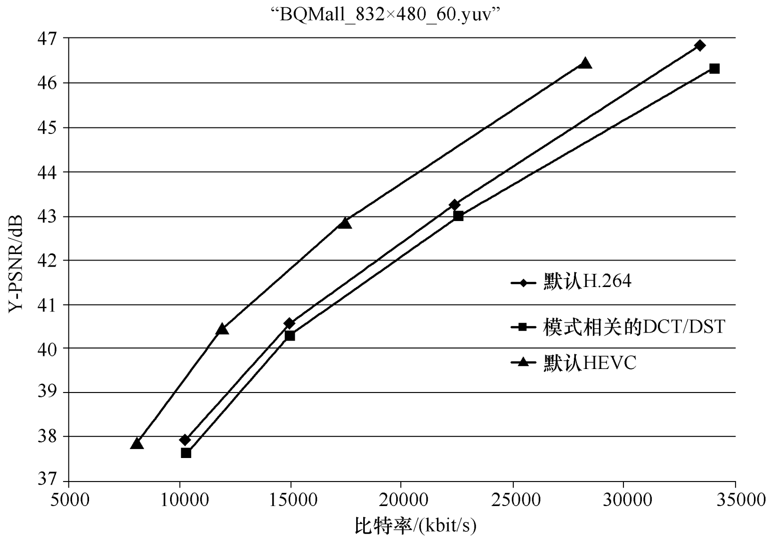


图 D. 6　BQMall 序列 Y-PSNR 随着比特率变化的曲线

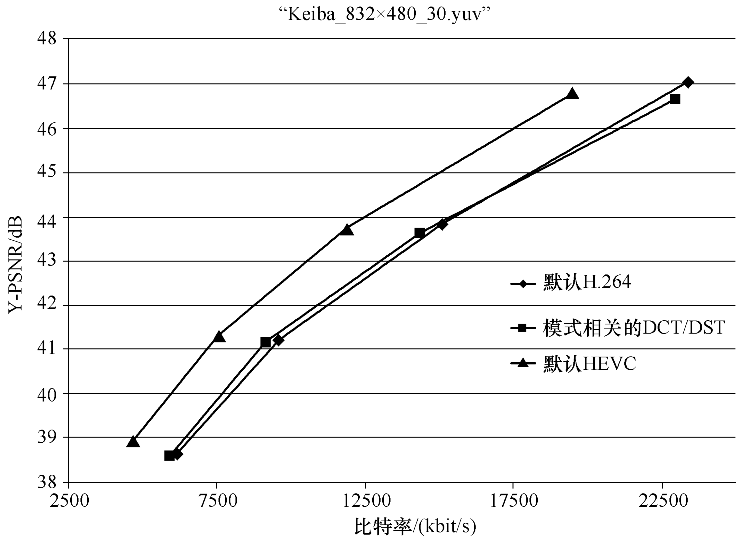


图 D. 7　Keiba 序列 Y-PSNR 随着比特率变化的曲线

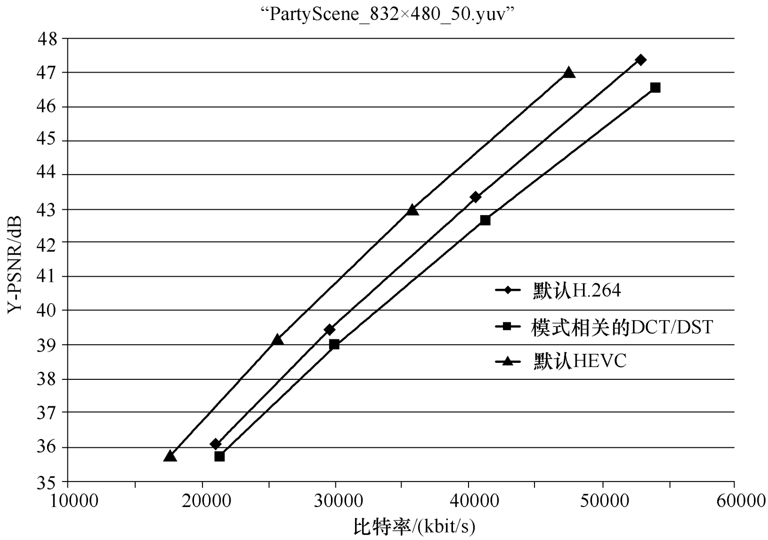


图 D. 8　PartyScent 序列 Y-PSNR 随比特率的变化曲线

D. 5. 3　HD（1920 × 1080）序列的测试结果

HD（1920 × 1080）序列的测试结果分别见表 D. 11 ~ 表 D. 14 和图 D. 9 ~ 图 D. 11。

表 D. 11　三个 1920 × 1080 序列的比特率和 PSNR 的比较

（带有 DCT/DST 的 H. 264/AVC 与默认 H. 264/AVC 相比）

序 列 名 称	QP	默认 H. 264/AVC 比特率 Y-PSNRU-PSNRV-PSNR (kbit/s) (dB)	带有 DCT/DST 的 H. 264/AVC 比特率 Y-PSNRU-PSNRV-PSNR (kbit/s) (dB)
BQTerrace	16	180787. 6847. 91846. 75147. 157	186144. 7247. 18846. 75147. 157
	20	129416. 8843. 88643. 90645. 222	132517. 4443. 32243. 90645. 222
	24	86247. 3640. 01641. 80243. 543	87560. 6439. 66541. 80243. 543
	28	55845. 6036. 87340. 04242. 078	56470. 8036. 61040. 04242. 078
Cactus	16	181654. 0847. 06645. 91246. 730	183171. 1246. 41945. 91246. 730
	20	115491. 3642. 57842. 64444. 690	115657. 2042. 30342. 64444. 690
	24	65324. 8839. 47640. 41442. 913	64902. 2439. 39040. 41442. 913
	28	39115. 9237. 38939. 03941. 249	38875. 6837. 36039. 03941. 249
Tennis	16	104600. 1646. 51346. 91548. 134	103009. 2046. 18346. 91548. 134
	20	55239. 8442. 99445. 37846. 868	54156. 4842. 91945. 37846. 868
	24	28356. 9641. 27544. 12445. 431	27876. 9641. 24344. 12445. 431
	28	17067. 1239. 59242. 90643. 879	16907. 2839. 53442. 90643. 879

表 D. 12　三个 1920 × 1080 序列的 BD-PSNR 和 BD-比特率的比较
（带有 DCT/DST 的 H. 264/AVC 与默认 H. 264/AVC）

Sequencename	BD-PSNR/dB	BD-比特率（%）
BQTerrace	−0. 6369	7. 2692
Cactus	−0. 2102	3. 5532
Tennis	−0. 0290	−0. 4242

表 D. 13　三个 1920 × 1080 序列的比特率和 PSNR 的比较
（H. 264/AVC 与 HEVC）

序列名称	QP	默认 H. 264/AVC 比特率 Y-PSNRU-PSNRV-PSNR （kbit/s）（dB）	默认 HEVC 比特率 Y-PSNRU-PSNRV-PSNR （kbit/s）（dB）
BQTerrace	16	180787. 6847. 91846. 75147. 157	149503. 4447. 992946. 711547. 3451
	20	129416. 8843. 88643. 90645. 222	108491. 7644. 619944. 039745. 3542
	24	86247. 3640. 01641. 80243. 543	72256. 5640. 630342. 008743. 7107
	28	55845. 6036. 87340. 04242. 078	42464. 6437. 064840. 133442. 1350
Cactus	16	181654. 0847. 06645. 91246. 730	159421. 6846. 814745. 760246. 8177
	20	115491. 3642. 57842. 64444. 690	95585. 7642. 540342. 870744. 8581
	24	65324. 8839. 47640. 41442. 913	46632. 9639. 315840. 471843. 1484
	28	39115. 9237. 38939. 03941. 249	26866. 3237. 431039. 033641. 4139
Tennis	16	104600. 1646. 51346. 91548. 134	87396. 2446. 492947. 204748. 5307
	20	55239. 8442. 99445. 37846. 868	34297. 6842. 867345. 474947. 0398
	24	28356. 9641. 27544. 12445. 431	18642. 4841. 534044. 320745. 6447
	28	17067. 1239. 59242. 90643. 879	11315. 2840. 227443. 023744. 0927

表 D. 14　三个 1920 × 1080 序列的 BD-PSNR 和 BD-比特率的比较
（H. 264/AVC 与 HEVC）

Sequencename	BD-PSNR/dB	BD-比特率（%）
BQTerrace	2. 1567	−21. 9681
Cactus	1. 3025	−19. 1702
Tennis	1. 4856	−33. 1268

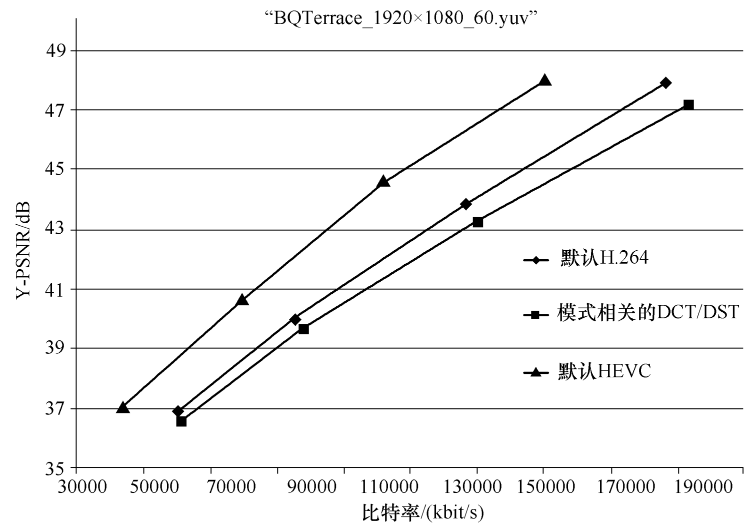


图 D. 9　BQTerrace 序列 Y-PSNR 随比特率的变化曲线

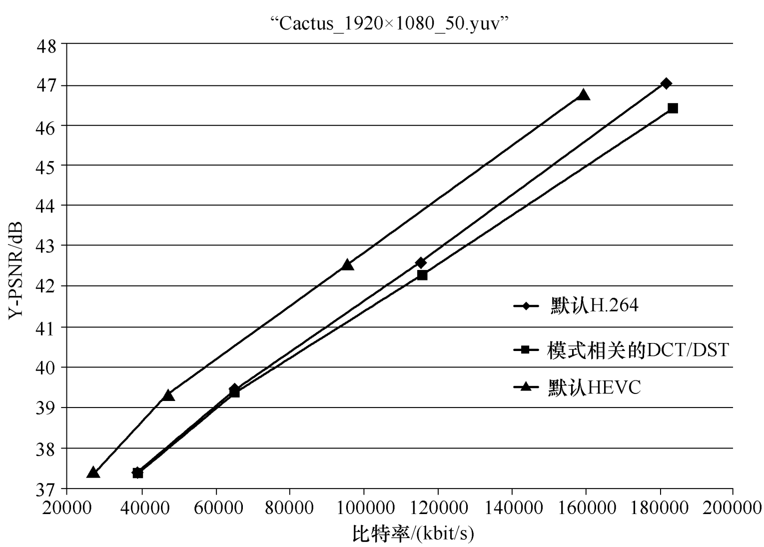


图 D. 10　cacuts 序列 Y-PSNR 随比特率的变化曲线

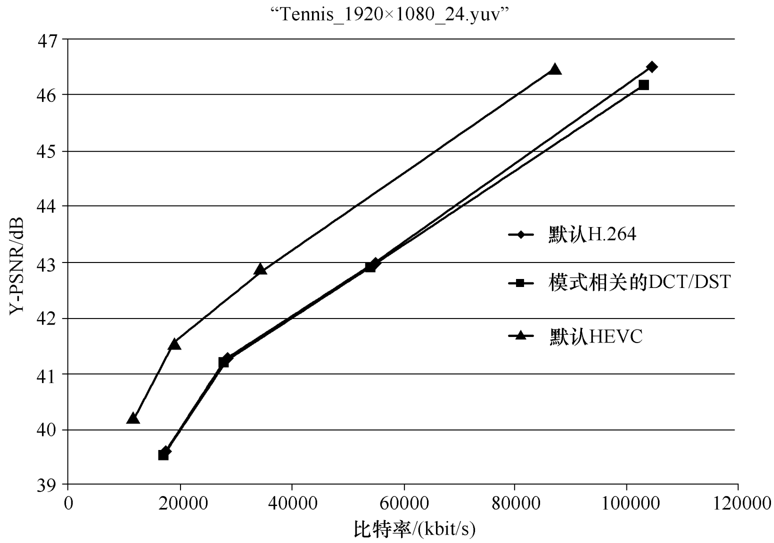


图 D. 11　tennis 序列 Y-PSNR 随比特率的变化曲线

D. 5. 4　高清（1080 × 720）序列的测试结果

高清（1080 × 720）序列的测试结果分别见表 D. 15 ~ 表 D. 18 和图 D. 12 ~ 图 D. 14。

表 D. 15　三个 1080 × 720 序列比特率和 PSNR 的比较

（带有 DCT/DST 的 H. 264/AVC 与默认 H. 264/AVC）

序列名称	QP	默认 H. 264/AVC 比特率 Y-PSNRU-PSNRV-PSNR (kbit/s) (dB)	带有 DCT/DST 的 H. 264/AVC 比特率 Y-PSNRU-PSNRV-PSNR (kbit/s) (dB)
Vidyo1	16	35621. 0447. 79048. 74549. 461	35020. 8047. 61548. 74549. 461
	20	21832. 0844. 88447. 69748. 595	21250. 0844. 77947. 69748. 595
	24	13267. 6842. 71146. 40447. 183	12959. 0442. 64046. 40447. 183
	28	8747. 2840. 59944. 96545. 540	8509. 4440. 50144. 96545. 540
Vidyo3	16	39600. 0047. 53649. 56449. 662	38542. 3247. 28949. 56449. 662
	20	24599. 2844. 25248. 88948. 689	23705. 2844. 15648. 88948. 689
	24	14207. 5242. 02347. 63047. 027	13705. 2042. 00447. 63047. 027
	28	8987. 5240. 04846. 40645. 260	8718. 2439. 96946. 40645. 260
Vidyo4	16	31649. 2848. 21549. 82150. 004	30944. 1648. 02249. 82150. 004
	20	19866. 0045. 40249. 18449. 379	19135. 2045. 27049. 18449. 379
	24	12456. 2443. 05747. 78347. 964	12089. 2842. 98047. 78347. 964
	28	8162. 8840. 82946. 10546. 256	7959. 3640. 65746. 10546. 256

表 D.16　三个 1080×720 序列的 BD-PSNR 和 BD-比特率的比较

（带有 DCT/DST 的 H.264/AVC 与默认 H.264/AVC）

序 列 名 称	BD-PSNR/dB	BD-比特率(%)
Vidyo1	0.0216	-0.5157
Vidyo3	0.0851	-1.8388
Vidyo4	0.0422	-0.8327

表 D.17　三个 1080×720 序列比特率和 PSNR 的比较（H.264/AVC 与 HEVC）

序列名称	QP	默认 H.264/AVC 比特率 Y-PSNRU-PSNRV-PSNR (kbit/s)(dB)	默认 HEVC 比特率 Y-PSNRU-PSNRV-PSNR (kbit/s)(dB)
Vidyo1	16	35621.0447.79048.74549.461	26735.7647.579949.206550.0380
	20	21832.0844.88447.69748.595	14940.9644.893847.811148.8625
	24	13267.6842.71146.40447.183	8738.8842.924146.568347.4347
	28	8747.2840.59944.96545.540	5621.5241.135945.123145.7768
Vidyo3	16	39600.0047.53649.56449.662	30262.0847.313550.330550.3281
	20	24599.2844.25248.88948.689	16935.3644.292649.066248.9256
	24	14207.5242.02347.63047.027	9007.4442.108747.699247.2448
	28	8987.5240.04846.40645.260	5832.9640.554746.410345.2869
Vidyo4	16	31649.2848.21549.82150.004	23430.2447.888650.454850.6957
	20	19866.0045.40249.18449.379	13652.6445.389749.350149.5508
	24	12456.2443.05747.78347.964	8428.0843.373147.847248.1619
	28	8162.8840.82946.10546.256	5282.8841.358746.196946.3486

表 D.18　三个 1080×720 序列 BD-PSNR 和 BD-比特率的比较

（H.264/AVC 与 HEVC）

序 列 名 称	BD-PSNR/dB	BD-比特率(%)
Vidyo1	2.0530	-33.5926
Vidyo3	1.9721	-33.2696
Vidyo4	2.0659	-33.2327

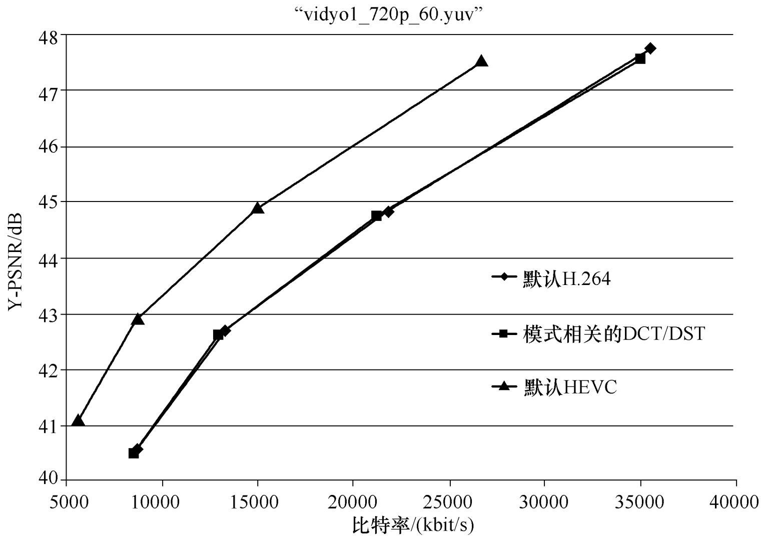


图 D. 12　vidyol 序列 Y-PSNR 随比特率的变化曲线

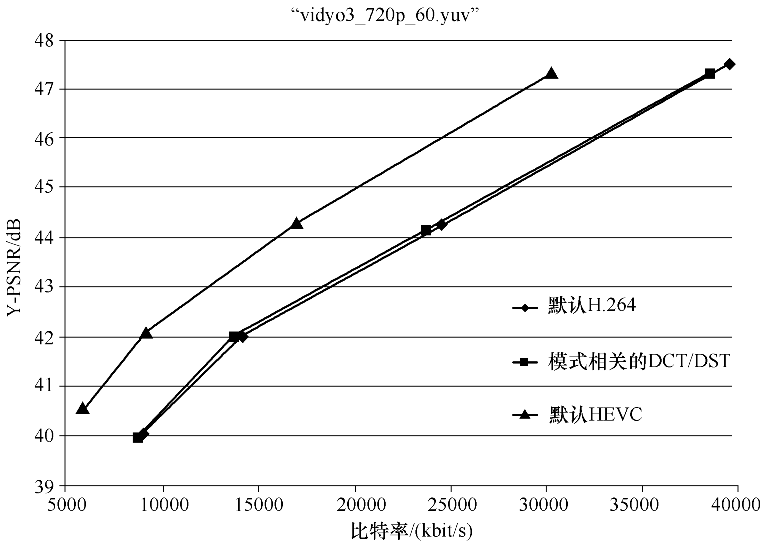


图 D. 13　vidyo3 序列 Y-PSNR 随比特率的变化曲线

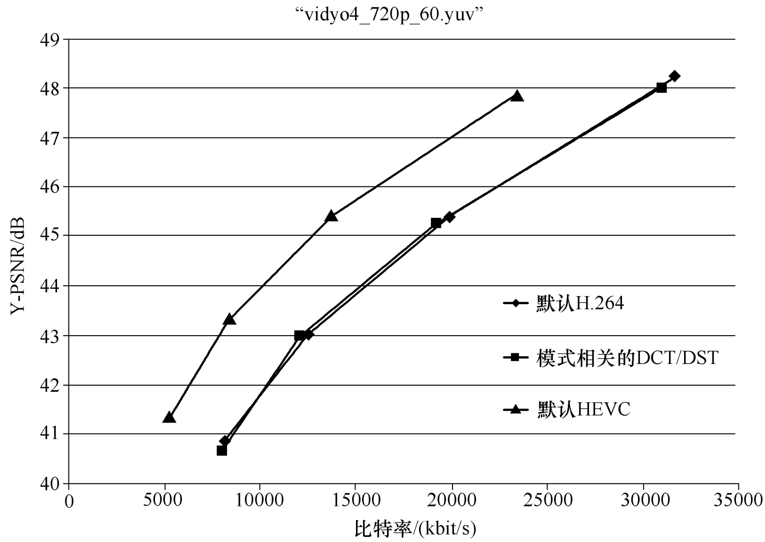


图 D. 14　vidyo4 序列 Y-PSNR 随比特率的变化曲线

D. 5. 5　用于 RaceHorse 序列 DCT/DST 不同组合的测试结果

为了观察到提出方案的性能，DCT/DST 的组合用于 9 种情况，从仅有水平模式和仅有垂直模式开始，然后扩展到第 1 类、第 2 类、第 1 和 2 类的组合、以及进一步扩展到全部所提方案中描述的类的组合，最后扩展到所有模式的 DST。测试结果列在表 D. 19 和表 D. 20 中。相应的曲线如图 D. 15 所示。

表 D. 19　DCT/DST 组合用于不同帧内预测模式的比特率和 PSNR 的比较

（默认 H. 264/AVC 与带有不同模式下 DCT/DST 的 H. 264/AVC）

用于帧内预测模式的 DST	QP	默认 H. 264/AVC 比特率 Y-PSNRU-PSNRV-PSNR （kbit/s）（dB）	带有 DCT/DST 的 H. 264/AVC 比特率 Y-PSNRU-PSNRV-PSNR （kbit/s）（dB）
DST 只用于模式 1	16	10105. 6847. 74447. 20147. 377	10100. 4047. 74847. 20147. 377
	20	7556. 1644. 08444. 03044. 497	7527. 3644. 05244. 03044. 497
	24	5429. 7640. 49641. 31341. 726	5435. 5240. 52441. 31341. 726
	28	3792. 2437. 09738. 89439. 324	3804. 4837. 17338. 89439. 324
DST 只用于模式 1 和 8	16	10105. 6847. 74447. 20147. 377	10151. 5247. 71047. 20147. 377
	20	7556. 1644. 08444. 03044. 497	7559. 5244. 02144. 03044. 497
	24	5429. 7640. 49641. 31341. 726	5439. 6040. 51041. 31341. 726
	28	3792. 2437. 09738. 89439. 324	3788. 8837. 14338. 89439. 324

（续）

用于帧内预测模式的 DST	QP	默认 H.264/AVC 比特率 Y-PSNRU-PSNRV-PSNR (kbit/s)(dB)	带有 DCT/DST 的 H.264/AVC 比特率 Y-PSNRU-PSNRV-PSNR (kbit/s)(dB)
DST 只用于模式 0	16	10105.6847.74447.20147.377	10119.8447.74647.20147.377
	20	7556.1644.08444.03044.497	7539.8444.09844.03044.497
	24	5429.7640.49641.31341.726	5423.5240.49741.31341.726
	28	3792.2437.09738.89439.324	3787.6837.14738.89439.324
DST 只用于模式 0,3,7	16	10105.6847.74447.20147.377	10193.2847.66447.20147.377
	20	7556.1644.08444.03044.497	7583.2844.49744.03044.497
	24	5429.7640.49641.31341.726	5422.8040.41341.31341.726
	28	3792.2437.09738.89439.324	3801.6037.11838.89439.324
DST 只用于模式 0,1,3,7	16	10105.6847.74447.20147.377	10175.5247.64247.20147.377
	20	7556.1644.08444.03044.497	7582.3243.99044.03044.497
	24	5429.7640.49641.31341.726	5423.2840.43841.31341.726
	28	3792.2437.09738.89439.324	3798.0037.12438.89439.324
DST 只用于模式 0,1,3,7,8	16	10105.6847.74447.20147.377	10238.6447.57147.20147.377
	20	7556.1644.08444.03044.497	7602.0043.94244.03044.497
	24	5429.7640.49641.31341.726	5453.7640.42041.31341.726
	28	3792.2437.09738.89439.324	3795.8437.12138.89439.324
DST 只用于模式 4,5,6	16	10105.6847.74447.20147.377	10291.6847.66347.20147.377
	20	7556.1644.08444.03044.497	7666.8043.94944.03044.497
	24	5429.7640.49641.31341.726	5508.4840.39341.31341.726
	28	3792.2437.09738.89439.324	3822.9637.05138.89439.324
模式相关的 DST	16	10105.6847.74447.20147.377	10306.3247.15547.20147.377
	20	7556.1644.08444.03044.497	7662.7243.61744.03044.497
	24	5429.7640.49641.31341.726	5476.0840.19841.31341.726
	28	3792.2437.09738.89439.324	3801.1236.93638.89439.324
DST 用于所有模式	16	10105.6847.74447.20147.377	18232.3244.35149.69849.648
	20	7556.1644.08444.03044.497	9331.6839.72744.47144.959
	24	5429.7640.49641.31341.726	5510.8837.40541.33041.779
	28	3792.2437.09738.89439.324	3755.5235.51038.89439.324

表 D. 20 BD-PSNR 与 BD-比特率

(默认 H. 264/AVC 与将 DCT/DST 用于不同帧内预测模式的 H. 264/AVC)

用于帧内预测模式的 DST	BD-PSNR/dB	BD-比特率(%)
DST 只用于模式 1	0. 0192	-0. 1817
DST 只用于模式 1 和 8	-0. 0277	0. 2627
DST 只用于模式 0	0. 0239	-0. 2468
DST 只用于模式 0,3,7	0. 0678	-0. 7543
DST 只用于模式 0,1,3,7	-0. 0844	0. 7942
DST 只用于模式 0,1,3,7,8	-0. 1595	1. 4838
DST 只用于模式 4,5,6	-0. 0236	0. 1937
模式相关的 DST	-0. 4913	4. 775
用于全部模式的 DST	-6. 3851	82. 9101

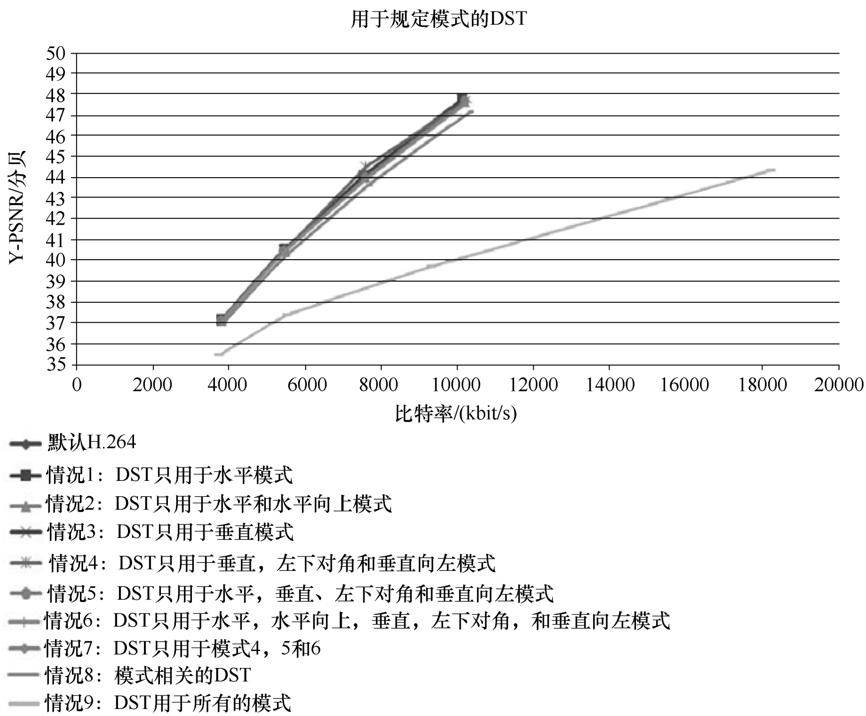


图 D. 15 DCT/DST 用于不同帧内预测模式 RaceHorses 序列的 Y-PSNR 随比特率的变化曲线

D. 6 结论与未来的工作

负的 BD-比特率或正的 BD-PSNR 表示较好的编码效率。HEVC 给出高于默认

H. 264 几乎 1. 5 ~2 分贝的性能改善。分析了 12 个输入视频序列中的 4 个，结果表明 H. 264 中模式相关的 DCT/DST 会提高某些性能。三个高清序列（1080 ×720）都给出了轻微的性能改善：低于 0. 05 分贝的 PSNR 增益和低于 1% 的节约比特率。当把模式相关的 DCT/DST 用于帧内预测模式的不同类别时，我们观察到只对水平和垂直模式有一点可见的轻微改善。

多数情况下性能的下降可能由于许多原因：

- 使用了非整数变换系数，导致了重建输出精确度的下降。
- H. 264 的预测模式只有 9 种，而 HEVC 有 33 种（角度模式）。在 H. 264 的情况下，DCT/DST 组合用于数量较少的预测方向上。
- 这里没有考虑最优预测模式的选择以及相应的模式相关的 DCT/DST。

为了实现大的性能改善，可以考虑许多其他因素。

- 性能取决于每块的帧内预测方向，可以改进变换系数的扫描顺序来获得更好的性能，而不是在 H. 264 中使用传统的 Z 型扫描。
- 可以使用率失真优化（RDO，Rate Distortion Optimization）来将模式相关的 DCT/DST 仅仅用于性能最好的模式。通过使用 H. 264 的高档次，较高分辨率的视频序列可用于分析目的。
- 还可以把分析推广到 4 ×4 亮度块以外的块，也可针对色度帧内预测残差展开分析。

附录 E H.264 的实现软件 JM，Intel IPP 和 x 264 的性能分析与比较

本附录基于 Santosh Kumar Muniyppa 的专题。

欲知详情，请参见 UTA 的 EE5359 课程网址（2010 秋季）http://www-ee.uta.edu/Dip/Courses/EE5359/index.html 上的专题。

H.264 的实现软件 JM，Intel IPP 和 x264 的性能分析与比较。

目标：实现 H.264 软件如 JM 软件，Intel IPP 和 x264 的性能分析。

动机：H.264 是多年来广泛接受的视频标准，而且出现了大量的实现软件，即 JM（Joint Model：联合模型）[H30]，Intel IPP（Integrated Performance Priminitives，集成性能原语）[X3]，x264 [X1]，FFmpeg（Fast Forward Moving Picture Experts Group，快速前向运动图像专家组）[X4]。因而性能分析专题将有助于选择一个针对您应用和需求的特定编译码器。

问题陈述：在此，我们使用不同的测试序列基于软件的性能展开对 H.264 软件 JM，Intel IPP 和 x264 的分析和比较。

E.1 H.264

H.264/AVC 是 ITU-T 视频编码专家组（VCEG，Video Coding Experts Group）和 ISO/IEC 运动图像专家组（MPEG，Moving Picture Experts Group）的最新的视频编码标准。H.264 是一个新的视频压缩方案，已经成为全球消费电子应用和个人电脑的数字视频标准。特别地，H.264 已经被选为下一代光盘格式的关键压缩方案（编译码器），即高清 DVD 和蓝光光盘（有时候叫作 BD 或 BD-ROM）。

H.264 已被 MPEG 采纳为面向数字媒体交换的 MPEG-4 中的视频压缩方案。H.264 有时候也被称为“MPEG-4 第 10 部分”（MPEG-4 规格的部分）或者叫作“AVC（MPEG-4 的高级视频编码）”。

H.264/AVC 在率失真效率方面实现了明显的提高，同时，与现存的标准如 MPEG-2 [S13] [H17] 视频相比，一般能节约两倍的比特率。

H.264 档次如图 E.1 所示。

H.264 的编码框图如图 E.2 所示。

H.264 的解码框图如图 E.3 所示。

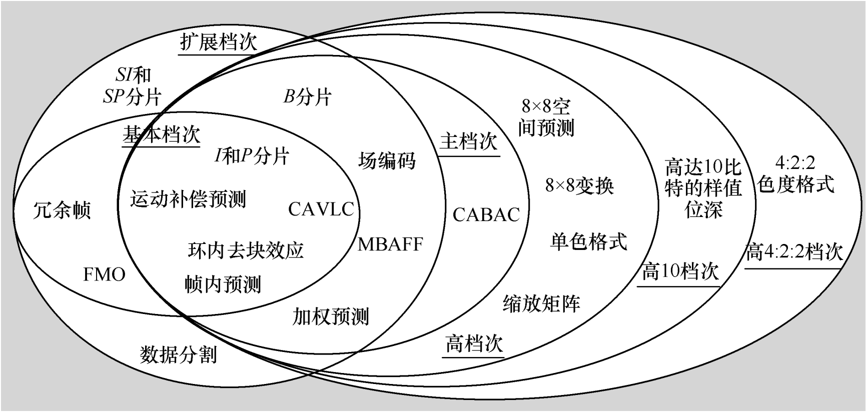


图 E. 1　H. 264 的档次［H25］（2006 IEEE）

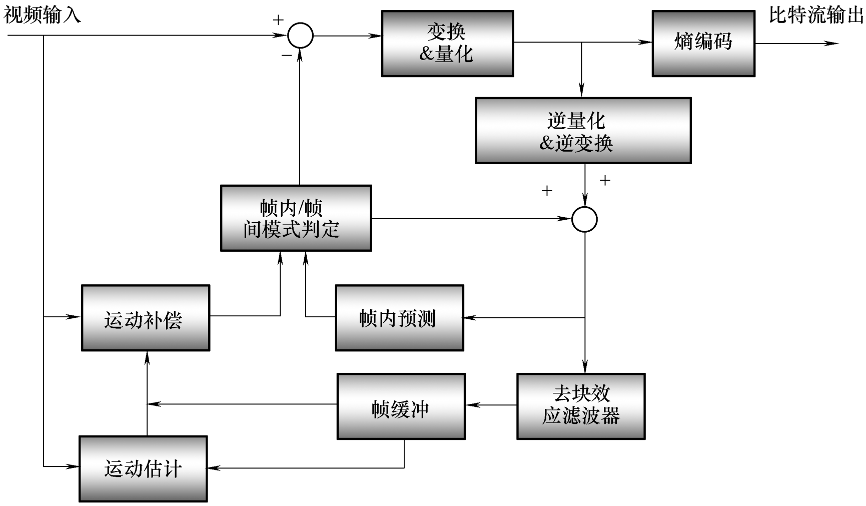


图 E. 2　H. 264 的编码框图［H25］（2006 IEEE）

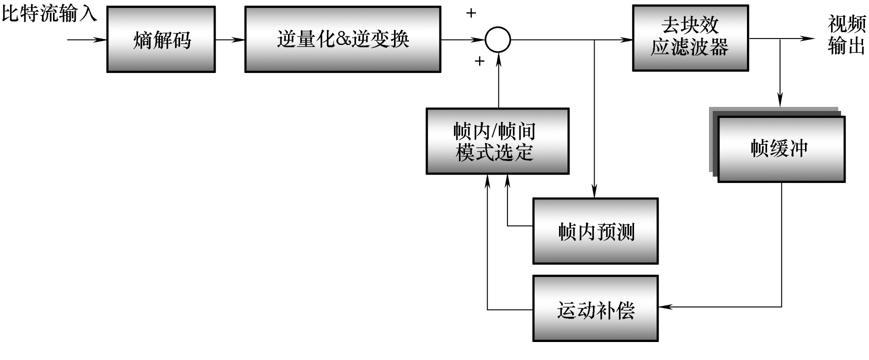


图 E. 3　H. 264 解码框图［H25］（2006 IEEE）

E.2 JM 软件

JM 软件是 ISO/IEC MPEG 联合视频小组（JVT）和 ITU-T 的 VCEG 的产品。

JM 软件的最新版本号是 18。既支持平面的又支持各行扫描的/打包的原始图像数据（即 yuv，rgb）。输入文件是一个配置文件（文本文件），文件中的部分合格参数如下：

- 输入文件；
- 待编码帧数；
- 输出帧宽度和高度；
- 档次、级别的选择；
- GOP 大小；
- 比特率控制。

E.3 x264

x264 的高性能归功于它的率控制、运动估计、宏块模式判定、量化和帧类型判定算法。此外，x264 使用了许多原始操作的组装优化代码。将 x264 与 JM 相比，平均起来，x264 比 JM 快 50 倍，且提供接近于 JM 的率失真性能［X2］。

它支持原始视频数据（yuv4mpeg 或只对 4:2:0 格式的 y4m），输出文件格式是.264，.mkv，mp4。输入通过命令提示来提供。可以提供的部分参数如下：

- 档次；
- 率控制；
- GOP 大小；
- 量化参数；
- 帧率。

E.4 Intel IPP

Intel 的集成性能原语（Intel IPP）是一个内容广泛面向多核高度优化的软件函数库，用于数字媒体和数据处理应用。Intel IPP 提供几千个优化的函数，涵盖了频繁使用的基本算法。Intel IPP 函数设计的交付性能超越了优化编译器单独能交付的性能［X5］。

1. 已知的限制

- 编码器假定输入视频数据对象包含了格式为 YUV420 的帧；
- 编码器不支持调整帧大小。因而输入和输出帧大小应该是相同的；
- 只支持主档次和高档次；

- 输入文件是 h264. par；
- 源文件；
- 待编码帧数；
- 帧率。

2. 目标

使用 QCIF 和 HDTV 视频测试序列，针对不同的档次和不同的比特率的详细分析将会展开。性能比较将基于这些参数进行：

- 编码和解码时间（s）；
- 压缩比；
- 峰值信噪比；
- 结构相似度指标［Q13］。

3. 所用的视频序列

- 基本档次和主档次，如图 E. 4 所示；



图 E. 4　akiyo_qcif. y4m

- 高档次，如图 E. 5 所示。



图 E. 5　sintel_720. y4m

4. CIF 和 QCIF 的 Y，Cb 和 Cr 的差别

CIF 和 QCIF 的 Y，Cb 和 Cr 的差别如图 E. 6 所示。

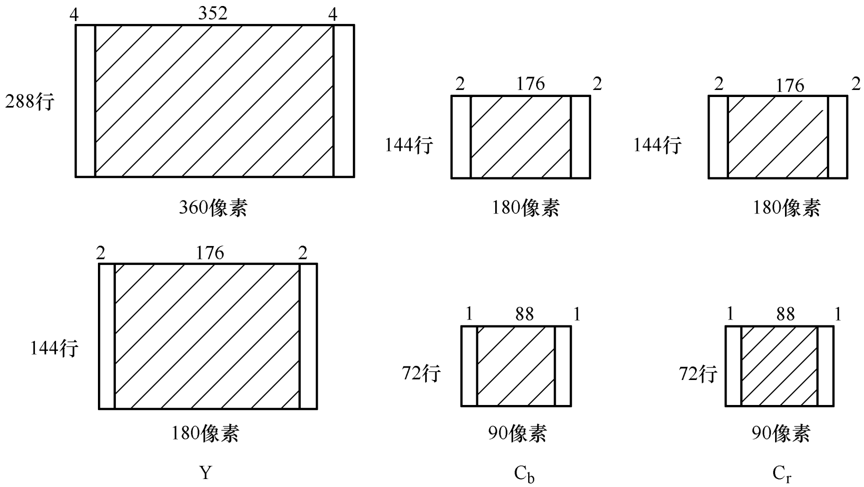


图 E. 6　CIF 和 QCIF 的 4∶2∶0 格式

E. 5　JM（17. 2）性能分析

1. 基本档次的 JM 性能

视频序列：akiyo_qcif. y4m

编码帧数：100

帧组：IBPBPBPBPB

量化参数：10，28，40，51

参考帧数：3（见图 E. 7 和表 E. 1）

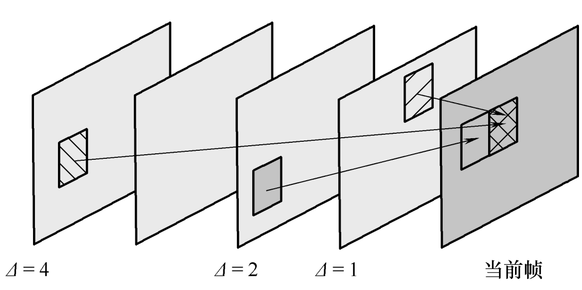


图 E. 7　运动估计的多个参考帧［H3］（2003 IEEE）

2. 主档次的 JM 性能

视频序列：akiyo_qcif. y4m

编码帧数：100

帧组：IBPBPBPBPB

量化参数：10，28，40，51

参考帧数：3（见表 E.2）

3. 高档次中的 JM 性能

视频序列：sintel_720p.y4m

编码帧数：100

帧组：IBPBPBPBPB

量化参数：10，28，40，51

参考帧数：3（见表 E.3）

SSIM、PSNR、编码时间、比特率、压缩比关于 QP（量化参数）的曲线图如图 E.8～E.12 所示。

表 E.1　基本档次的 JM 性能

量化参数	SSIM	PSNR/dB	编码时间/s	比特率/(kbit/s)	压缩比
10	0.9969	51.086	525.499	1237.16	7.367063
28	0.971	38.234	248.934	229.5	39.5
40	0.861	29.906	183.088	71.38	123.7667
51	0.5814	22.532	145.103	25.35	337.5455

表 E.2　主档次的 JM 性能

量化参数	SSIM	PSNR/dB	编码时间/s	比特率/(kbit/s)	压缩比
10	0.9975	51.832	38.428	1292.51	7.058935
28	0.9722	38.431	35.715	204.14	44.20238
40	0.8609	29.836	34.030	66.05	137.5185
51	0.3909	12.79	33.082	12.76	618.8333

表 E.3　高档次的 JM 性能

量化参数	SSIM	PSNR/dB	编码时间/s	比特率/(kbit/s)	压缩比
10	0.9989	63.115	1438.401	14044.44	23.85827
28	0.9796	47.424	1394.849	2001.91	167.3006
40	0.8692	37.834	1307.674	476.25	702.8351
51	0.5354	10.716	1059.323	109.25	3030

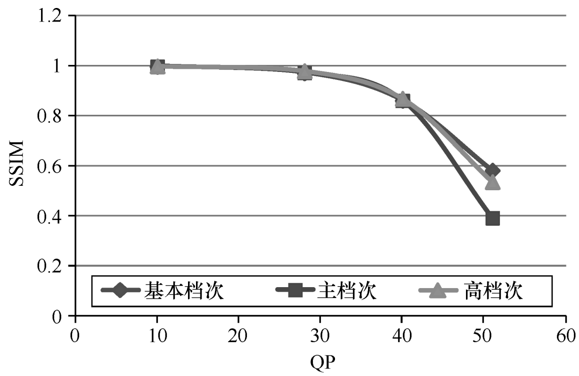


图 E. 8　基本档次、主档次和高档次的 SSIM 与 QP 的关系曲线

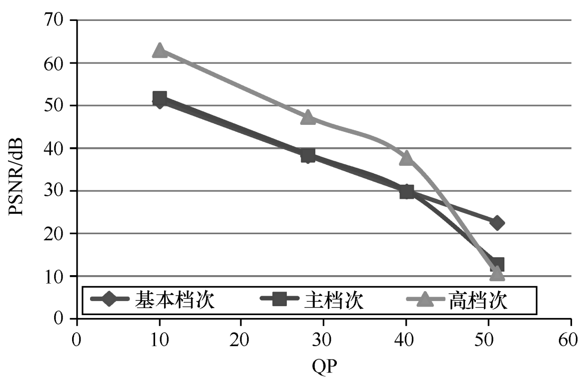


图 E. 9　基本档次、主档次和高档次的 PSNR 与 QP 的关系曲线

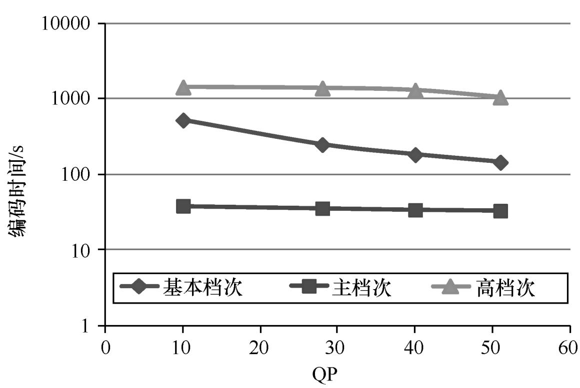


图 E. 10　基本档次、主档次和高档次的编码时间与 QP 的关系曲线

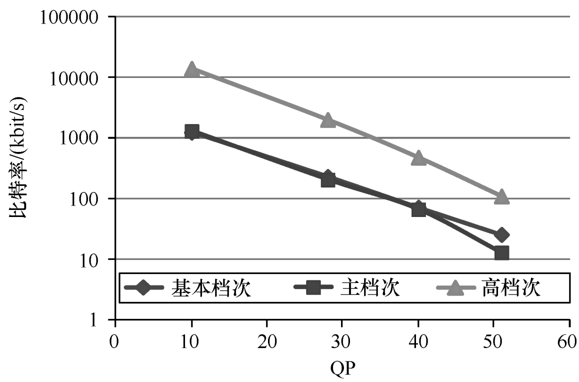


图 E. 11　基本档次、主档次和高档次的比特率与 QP 的关系曲线

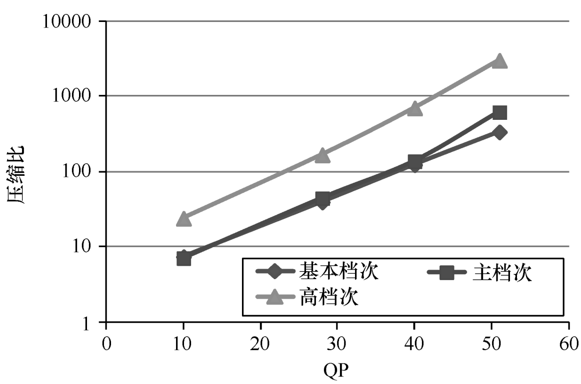


图 E. 12　基本档次、主档次和高档次的压缩比与量化参数的关系曲线

E. 6　x264 性能分析

1. 基本档次中 x264 的性能（见表 E. 4）

视频序列：akiyo_qcif. y4m

编码帧数：100

帧组：IBPBPBPBPB

量化参数：10，28，40，51

参考帧数：3

2. 主档次中 x264 的性能（见表 E. 5）

视频序列：akiyo_qcif. y4m

编码帧数：100

帧组：IBPBPBPBPB

量化参数：10，28，40，51

参考帧数：3

3. 高档次中 x264 性能（见表 E.6）

视频序列：sintel_720p. y4m

编码帧数：100

帧组：IBPBPBPBPB

量化参数：10，28，40，51

参考帧数：3

表 E.4　基本档次中 x264 的性能

量化参数	SSIM	PSNR/dB	编码时间/s	比特率/(kbit/s)	压缩比
10	0.997426	51.895	0.813	426.28	21.33908
28	0.977708	39.174	0.396	36.16	247.5333
40	0.9046	31.617	0.392	9.78	928.25
51	0.737023	25.914	0.271	5.35	1237.667

表 E.5　主要档次中 x264 的性能

量化参数	SSIM	PSNR/dB	编码时间/s	比特率/(kbit/s)	压缩比
10	0.997371	51.587	0.7949	288.15	31.4661
28	0.977456	39.302	0.5029	30.29	285.6154
40	0.904055	31.814	0.4889	11.28	742.6
51	0.724986	25.959	0.327	6.4	1237.667

表 E.6　高档次中 x264 的性能

量化参数	SSIM	PSNR/dB	编码时间/s	比特率/(kbit/s)	压缩比
10	0.998286	63.175	17.85714286	5465.59	61.30845
28	0.99331	55.436	7.849293564	510.28	655.5288
40	0.982018	42.128	4.830917874	144.31	2311.017
51	0.959803	35.188	4.636068614	50.81	6492.857

SSIM、PSNR、编码时间、比特率、压缩比与量化参数（QP）的关系曲线如图 E.13 ~ 图 E.17 所示。

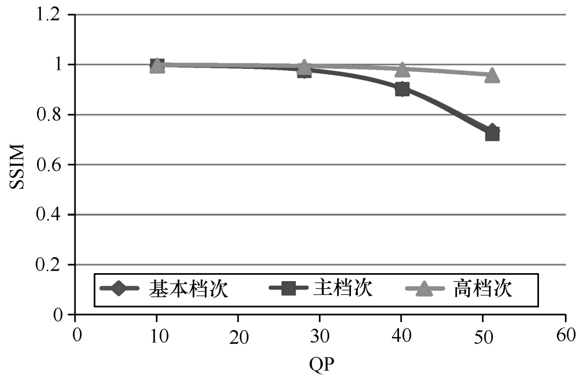


图 E.13　基本档次、主档次和高档次 SSIM 与 QP 的关系曲线

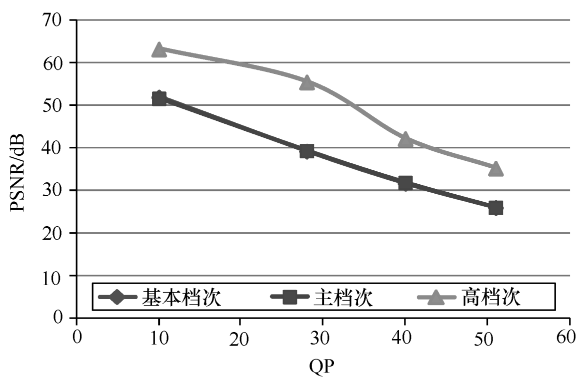


图 E.14　基本档次、主档次和高档次 PSNR 与 QP 的关系曲线

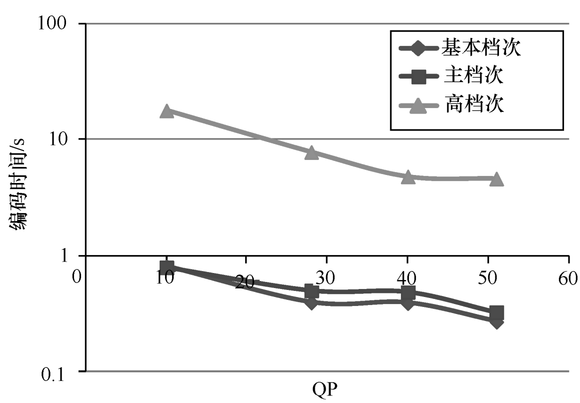


图 E.15　基本档次、主档次和高档次编码时间与 QP 的关系曲线

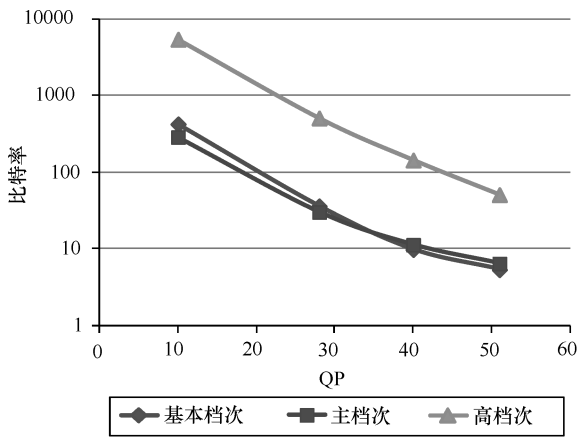


图 E.16 基本档次、主档次和高档次比特率与 QP 的关系曲线

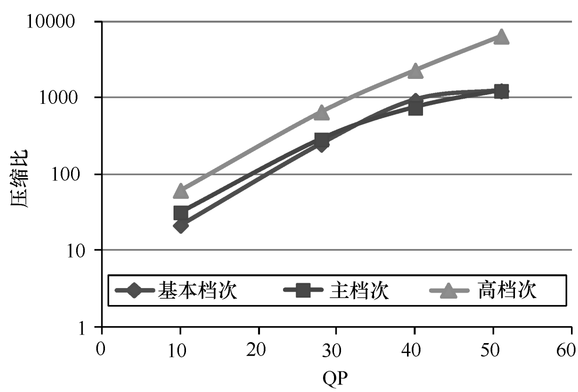


图 E.17 基本档次、主档次和高档次压缩比与 QP 的关系曲线

E.7 Intel IPP 性能分析

1. 基本档次中 Intel IPP 的性能（见表 E.7）

视频序列：akiyo_qcif.y4m

编码帧数：100

帧组：IBPBPBPBPB

比特率（kbit/s）：1000，100，50，10

参考帧数：3

2. 主档次中 Intel IPP 的性能（见表 E.8）

视频序列：akiyo_qcif.y4m

编码帧数：100

帧组：IBPBPBPBPB

比特率（kbit/s）：1000，100，50，10

参考帧数：3

3. 高档次中 Intel IPP 的性能（见表 E.9）

视频序列：sintel_720p.y4m

编码帧数：100

帧组：IBPBPBPBPB

比特率（kbit/s）：1000，100，50，10

参考帧数：3

SSIM，PSNR，编码时间，压缩比关于比特率的关系曲线如图 E.18 ~ 图 E.21 所示。

表 E.7　基本档次中 IntelIPP 的性能

比特率(kbit/s)	SSIM	PSNR/dB	总的编码时间/s	压缩比
1000	0.9963	49.8634	0.66	8.47
100	0.979	41.2845	0.45	68.57
50	0.9392	35.8397	0.38	137.44
10	0.6553	24.6113	0.35	154.37

表 E.8　主档次中 IntelIPP 的性能

比特率(kbit/s)	SSIM	PSNR/dB	总的编码时间/s	压缩比
1000	0.9962	49.1608	0.69	9.4
100	0.983	41.1057	0.44	77.68
50	0.9492	35.3562	0.4	138.97
10	0.655	24.6113	0.39	159.6

表 E.9　高档次中的 IntelIPP 性能

比特率(kbit/s)	SSIM	PSNR/dB	总的编码时间/s	压缩比
1000	0.998286	61.755	11.71	366.75
100	0.98331	55.8701	4.76	3317.02
50	0.942018	42.128	4.2	5153.59
10	0.899803	35.188	4.17	5856.63

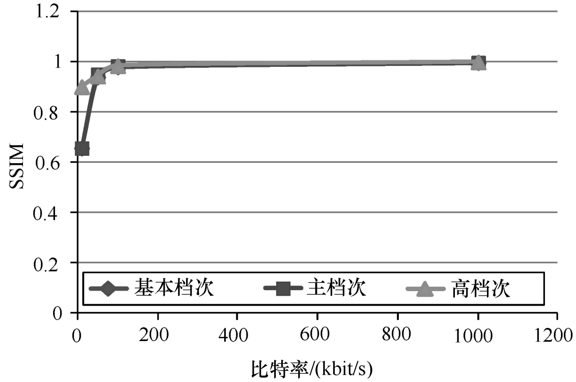


图 E. 18　基本档次、主档次和高档次中 SSIM 与比特率的关系曲线

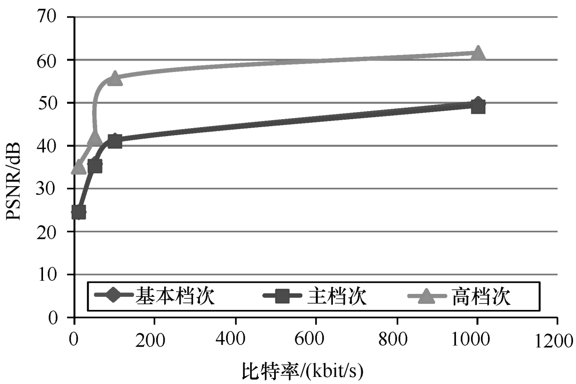


图 E. 19　基本档次、主档次和高档次中 PSNR 与比特率的关系曲线

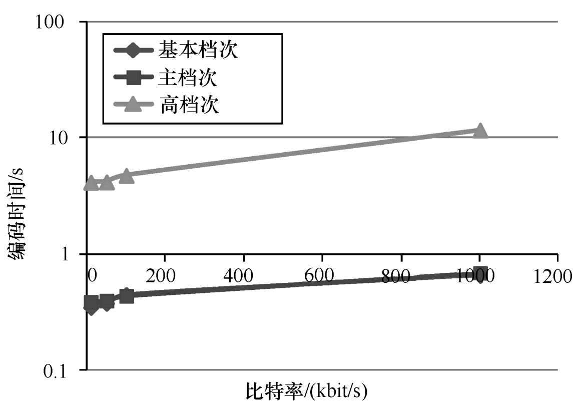


图 E. 20　基本档次、主档次和高档次中编码时间与比特率的关系曲线

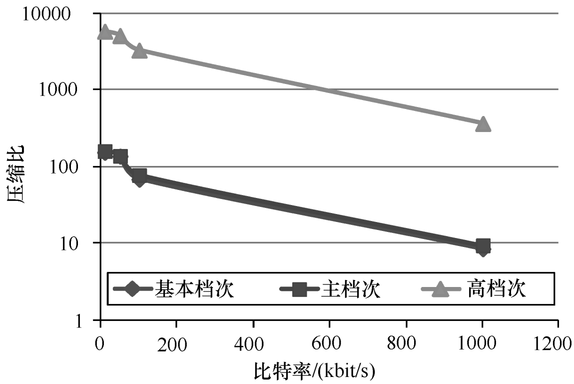


图 E. 21　基本档次、主档次和高档次中压缩比与比特率的关系曲线

E. 8　在基本档次、主档次和高档次中 JM、x264 和 Intel IPP 软件的 SSIM 比较

基本档次中 JM、x264 和 Intel IPP 软件的 SSIM 比较（akiyo_qcif. y4m）如表 E. 10 和图 E. 22 所示。

表 E. 10　基本档次中 JM、x264 和 Intel IPP 软件的 SSIM 比较（akiyo_qcif. y4m）

JM		x264		IntelIPP	
比特率/(kbit/s)	SSIM	比特率/(kbit/s)	SSIM	比特率/(kbit/s)	SSIM
1237. 16	0. 9969	642. 01	0. 997426	1000	0. 9963
229. 5	0. 971	99. 08	0. 977708	100	0. 979
71. 38	0. 861	34. 3	0. 9046	50	0. 9392
25. 35	0. 5814	15. 55	0. 737023	10	0. 6553

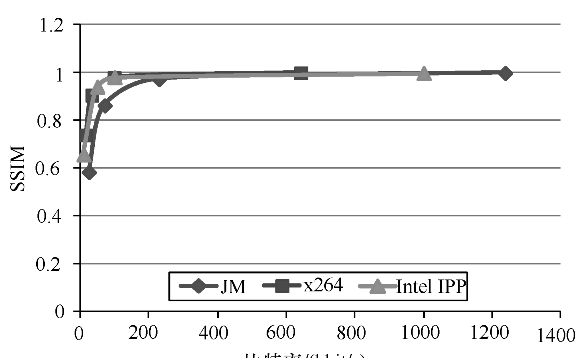


图 E. 22　基本档次中 JM、x264 和 Intel IPP 软件实现的 SSIM 与比特率的关系曲线（akiyo_qcif. y4m）

主档次中 JM、x264 和 Intel IPP 软件的 SSIM 比较（akiyo_qcif. y4m）如表 E. 11 和图 E. 23 所示。

表 E. 11　主档次中 JM、x264 和 Intel IPP 软件实现的 SSIM（akiyo_qcif. y4m）

JM		x264		IntelIPP	
比特率/(kbit/s)	SSIM	比特率/(kbit/s)	SSIM	比特率/(kbit/s)	SSIM
1292. 51	0. 9975	558. 28	0. 997588	1000	0. 9962
204. 14	0. 9722	93. 3	0. 979848	100	0. 983
66. 05	0. 8609	33. 38	0. 909762	50	0. 9492
12. 76	0. 3909	14. 97	0. 731834	10	0. 655

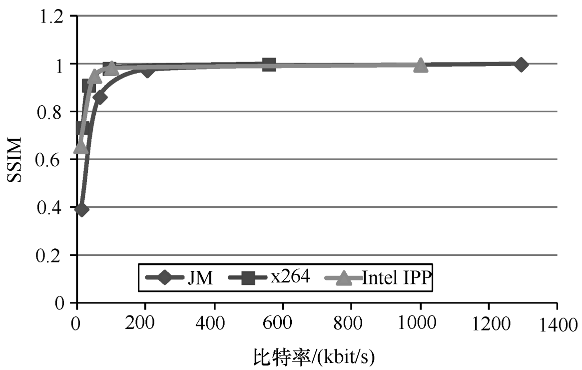


图 E. 23　主档次中 JM、x264 和 Intel IPP 实现的 SSIM 与比特率的关系曲线（akiyo_qcif. y4m）

高档次中 JM、x264 和 Intel IPP 软件的 SSIM 比较（sintel_780p. y4m）如表 E. 12 和图 E. 24 所示。

表 E. 12　高档次中 JM、x264 和 Intel IPP 软件 SSIM 的比较（sintel_780p. y4m）

JM		x264		IntelIPP	
比特率/(kbit/s)	SSIM	比特率/(kbit/s)	SSIM	比特率/(kbit/s)	SSIM
14044. 44	0. 9989	5465. 59	0. 998286	1000	0. 998286
2001. 91	0. 9796	510. 28	0. 99331	100	0. 98331
476. 25	0. 8692	144. 31	0. 982018	50	0. 942018
109. 25	0. 5354	50. 81	0. 959803	10	0. 899803

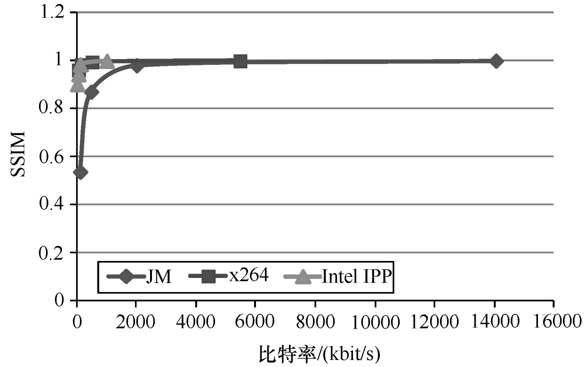


图 E. 24　高档次中 JM、x264 和 Intel IPP 实现的 SSIM 与比特率的关系曲线（sintel_780p. y4m）

E. 9　基本档次、主档次和高档次中 JM、x264 和 Intel IPP 软件实现的 PSNR 对比

基本档次中 JM、x264 和 Intel IPP 软件实现的 PSNR 对比（akiyo_qcif. y4m）如表 E. 13 和图 E. 25 所示。

表 E. 13　基本档次中 JM、x264 和 Intel IPP 实现的 PSNR 对比（akiyo_qcif. y4m）

JM		x264		IntelIPP	
比特率/(kbit/s)	PSNR/dB	比特率/(kbit/s)	PSNR/dB	比特率/(kbit/s)	PSNR/dB
1237. 16	51. 086	642. 01	52. 216	1000	49. 8634
229. 5	38. 234	99. 08	40. 082	100	41. 2845
71. 38	29. 906	34. 3	32. 173	50	35. 8397
25. 35	22. 532	15. 55	26. 25	10	24. 6113

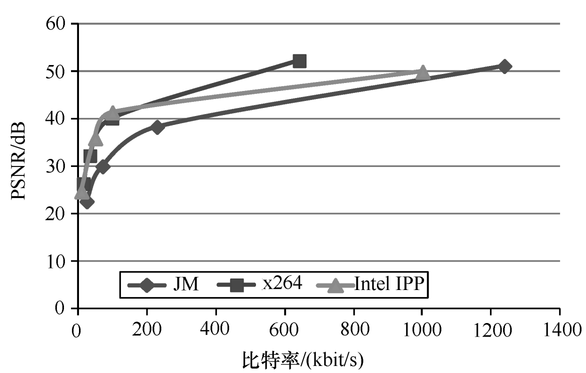


图 E. 25　基本档次中 JM、x264 和 Intel IPP 实现的 PSNR 与比特率的关系曲线（akiyo_qcif. y4m）

主档次中 JM、x264 和 Intel IPP 软件实现的 PSNR 对比（akiyo_qcif. y4m）如表 E. 14 和图 E. 26 所示。

表 E. 14 主档次中 JM、x264 和 Intel IPP 软件实现的 PSNR 对比（akiyo_qcif. y4m）

JM		x264		IntelIPP	
比特率/(kbit/s)	PSNR/dB	比特率/(kbit/s)	PSNR/dB	比特率/(kbit/s)	PSNR/dB
1292. 51	51. 832	558. 28	52. 162	1000	49. 1608
204. 14	38. 431	93. 3	40. 115	100	41. 1057
66. 05	29. 836	33. 38	32. 222	50	35. 3562
12. 76	12. 79	14. 97	26. 171	10	24. 6113

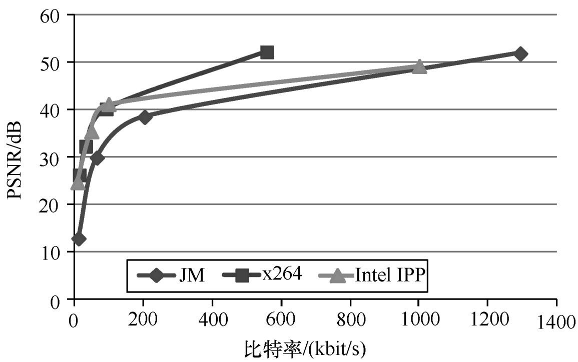


图 E. 26 主档次中 JM、x264 和 Intel IPP 实现的 PSNR 与比特率的关系曲线（akiyo_qcif. y4m）

高档次中 JM、x264 和 Intel IPP 软件实现的 PSNR 对比（sintel_780p. y4m）如表 E. 15 和图 E. 27 所示。

表 E. 15 高档次中 JM、x264 和 Intel IPP 软件实现的 PSNR 对比（sintel_780p. y4m）

JM		x264		IntelIPP	
比特率/(kbit/s)	PSNR/dB	比特率/(kbit/s)	PSNR/dB	比特率/(kbit/s)	PSNR/dB
14044. 44	63. 115	5465. 59	63. 175	1000	61. 755
2001. 91	47. 424	510. 28	55. 436	100	55. 8701
476. 25	37. 834	144. 31	42. 128	50	42. 128
109. 25	10. 716	50. 81	35. 188	10	35. 188

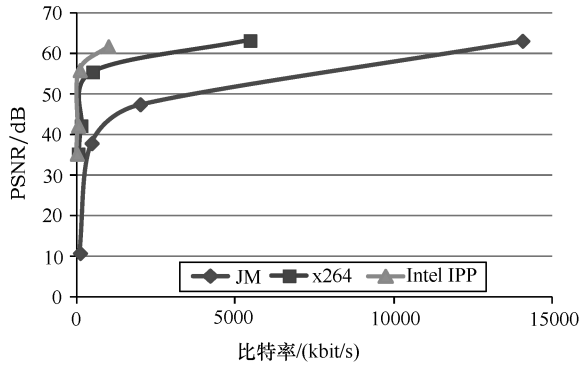


图 E. 27　高档次中 JM、x264 和 Intel IPP 实现的 PSNR 与比特率的关系曲线（sintel_780p. y4m）

E. 10　基本档次、主档次和高档次中 JM、x264 和 Intel IPP 软件所用编码时间的对比

基本档次中 JM、x264 和 Intel IPP 软件所用编码时间的对比（akiyo_qcif. y4 m）如表 E. 16 和图 E. 28 所示。

表 E. 16　基本档次中 JM、x264 和 Intel IPP 软件所用的编码时间比较（akiyo_qcif. y4m）

JM		x264		IntelIPP	
比特率/(kbit/s)	编码时间/s	比特率/(kbit/s)	编码时间/s	比特率/(kbit/s)	编码时间/s
1237. 16	525. 499	642. 01	0. 564016	1000	0. 66
229. 5	248. 934	99. 08	0. 325998	100	0. 45
71. 38	183. 088	34. 3	0. 197707	50	0. 38
25. 35	145. 103	15. 55	0. 163999	10	0. 35

主档次中 JM、x264 和 Intel IPP 软件实现的编码时间比较（akiyo_qcif. y4m）如表 E. 17 和图 E. 29 所示。

表 E. 17　主档次中 JM、x264 和 Intel IPP 编码时间的对比（akiyo_qcif. y4m）

JM		x264		IntelIPP	
比特率/(kbit/s)	编码时间/s	比特率/(kbit/s)	编码时间/s	比特率/(kbit/s)	编码时间/s
1292. 51	38. 428	558. 28	0. 843028	1000	0. 69
204. 14	35. 715	93. 3	0. 396999	100	0. 44
66. 05	34. 03	33. 38	0. 325998	50	0. 4
12. 76	33. 082	14. 97	0. 281001	10	0. 39

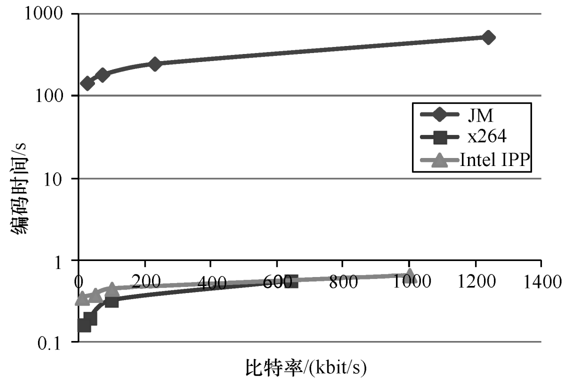


图 E. 28 基本档次中 JM、x264 和 Intel IPP 实现的编码时间与比特率的关系曲线（akiyo_qcif. y4m）

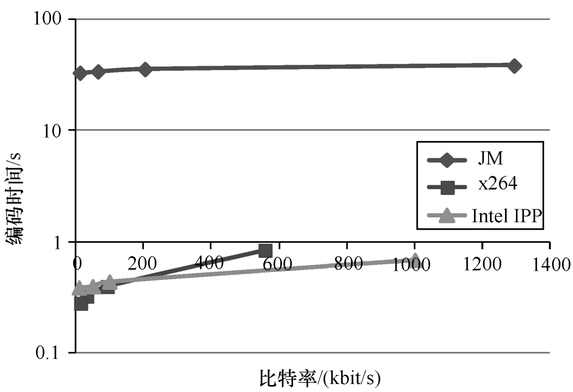


图 E. 29 主档次中 JM、x264 和 Intel IPP 实现的编码时间与比特率的关系曲线（akiyo_qcif. y4m）

高档次中 JM、x264 和 Intel IPP 软件实现的编码时间比较（sintel_780p. y4m）如表 E. 18 和图 E. 30 所示。

表 E. 18 高档次中 JM、x264 和 Intel IPP 软件的编码时间对比（sintel_780p. y4m）

JM		x264		IntelIPP	
比特率/(kbit/s)	编码时间/s	比特率/(kbit/s)	编码时间/s	比特率/(kbit/s)	编码时间/s
14044. 44	1438. 401	5465. 59	17. 85714	1000	11. 71
2001. 91	1394. 849	510. 28	7. 849294	100	4. 76
476. 25	1307. 674	144. 31	4. 830918	50	4. 2
109. 25	1059. 323	50. 81	4. 636069	10	4. 17

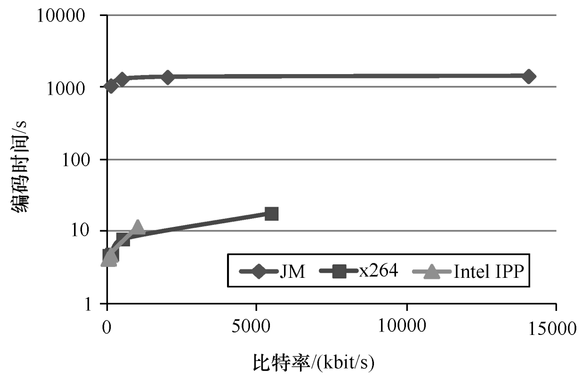


图 E. 30　高档次中 JM、x264 和 Intel IPP 实现的编码时间与比特率的关系曲线（sintel_780p. y4m）

E. 11　基本档次、主档次和高档次中 JM、x264 和 Intel IPP 软件实现的压缩比对比

基本档次中 JM、x264 和 Intel IPP 软件实现的压缩比对比（akiyo_qcif. y4m）如表 E. 19 和图 E. 31 所示。

表 E. 19　基本档次中 JM、x264 和 Intel IPP 压缩比的对比（akiyo_qcif. y4m）

JM		x264		Intel IPP	
比特率/(kbit/s)	压缩比	比特率/(kbit/s)	压缩比	比特率/(kbit/s)	压缩比
1237. 16	7. 367063	642. 01	14. 17176	1000	8. 47
229. 5	39. 5	99. 08	90. 56098	100	68. 57
71. 38	123. 7667	34. 3	265. 2143	50	137. 44
25. 35	337. 5455	15. 55	530. 4286	10	154. 37

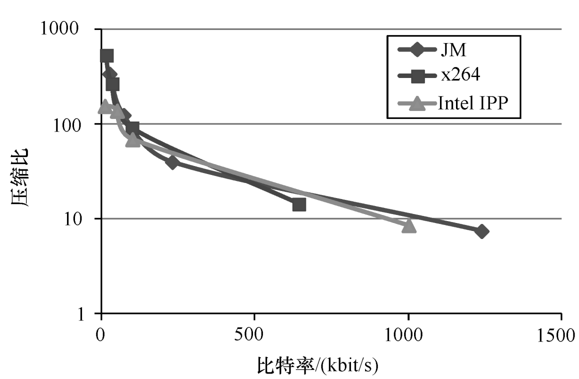


图 E. 31　基本档次中 JM、x264 和 Intel IPP 的压缩比与比特率的关系曲线（akiyo_qcif. y4m）

主档次中 JM、x264 和 Intel IPP 软件实现的压缩比对比（akiyo_qcif. y4m）如表 E. 20 和图 E. 32 所示。

表 E. 20　主档次中 JM、x264 和 Intel IPP 软件压缩比对比（akiyo_qcif. y4m）

JM		x264		Intel IPP	
比特率/(kbit/s)	压缩比	比特率/(kbit/s)	压缩比	比特率/(kbit/s)	压缩比
1292. 51	7. 058935	558. 28	16. 28509	1000	9. 4
204. 14	44. 20238	93. 3	97. 71053	100	77. 68
66. 05	137. 5185	33. 38	265. 2143	50	138. 97
12. 76	618. 8333	14. 97	530. 4286	10	159. 6

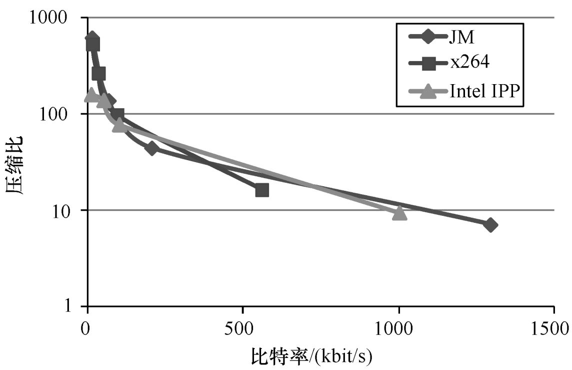


图 E. 32　主档次中 JM、x264 和 Intel IPP 实现的压缩比与比特率关系曲线（akiyo_qcif. y4m）

高档次中 JM、x264 和 Intel IPP 软件的压缩比对比（sintel_780p. y4m）如表 E. 21 和图 E. 33 所示。

表 E. 21　高档次中 JM、x264 和 Intel IPP 软件实现的压缩比对比（sintel_780p. y4m）

JM		x264		IntelIPP	
比特率/(kbit/s)	压缩比	比特率/(kbit/s)	压缩比	比特率/(kbit/s)	压缩比
14044. 44	23. 85827	5465. 59	61. 30845	1000	366. 75
2001. 91	167. 3006	510. 28	655. 5288	100	3317. 02
476. 25	702. 8351	144. 31	2311. 017	50	5153. 59
109. 25	3030	50. 81	6492. 857	10	5856. 63

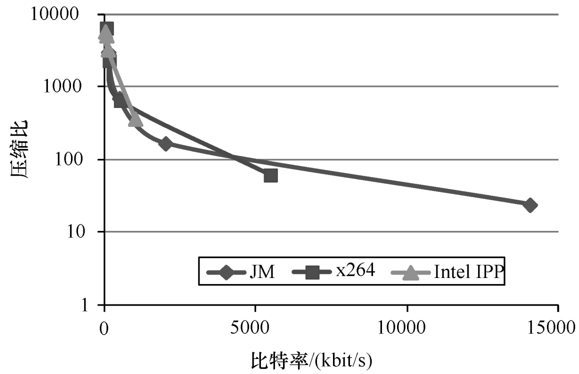


图 E. 33　高档次中 JM、x264 和 Intel IPP 实现的压缩比与比特率关系曲线（sintel_780p. y4m）

E. 12　结论

档　　次	基本档次	主　档　次	高　档　次
SSIM	x264 和 Intel IPP 都提供好于 JM 的结果	x264 和 Intel IPP 都提供好于 JM 的结果	x264 和 Intel IPP 都提供好于 JM 的结果
PSNR	x264 和 Intel IPP 都胜过 JM。x264 略优于 Intel IPP	x264 和 Intel IPP 都胜过 JM。x264 略优于 Intel IPP	x264 和 Intel IPP 都胜过 JM。Intel IPP 略优于 x264
编码时间	x264 和 Intel IPP 都胜过 JM。x264 略优于 Intel IPP	x264 和 Intel IPP 都胜过 JM。x264 略优于 Intel IPP	x264 和 Intel IPP 都胜过 JM
压缩比	x264 和 Intel IPP 都胜过 JM。x264 略优于 Intel IPP	x264 和 Intel IPP 都胜过 JM。x264 略优于 Intel IPP	x264 和 Intel IPP 都胜过 JM。Intel IPP 略优于 x264

E. 13　未来的工作

- 针对不同的视频序列进行分析；
- 针对不同的视频格式即 4：2：2，4：4：4 进行分析；
- 针对不同的视频分辨率即 352 ×288、480p、1080p 进行分析。

附录 F　基于 H.264 “只编码 I 帧[㊀]” 的 AIC 实现以及与其他静止帧图像编码标准如 JPEG、JPEG 2000、JPEG-LS 和 JPEG-XR 的比较

本附录基于 Radhika Veerla 的专题。

欲知详情，请参见 UTA 的 EE5359 课程网址 http://www-ee.uta.edu/Dip/Courses/EE5359/index.html 上的专题（2008 年夏季）

F.1　前言

基于“只编码 I 帧”利用微软 Visual Studio，我们实现了高级图像编码（AIC，Advance Image Coding），并且与其他图像压缩技术（如 JPEG、JPEG2000、JPEG-LS、JPEG-XR）和 H.264 对比了结果。在各种测试图像上进行了编码仿真。实验结果用比特率、质量-PSNR 来描述。本专题只考虑 H.264/AVC 帧内编码的主档次和高（FRExt）档次，JPEG 使用基本方法并让所有的编译码器都进行有损压缩。

AIC 的目的是［AC1］以降低的复杂度提供更好的质量，同时优化可读性和简明性。尽管其目的不在于优化速度，但它比许多 JPEG2000 编译码器［J8］运行更快。因为它的高质量图像，AIC 的应用包括了医学成像。H.264 技术旨在以相当低的比特、适当的复杂度提供好的视频质量，同时为范围广阔的应用提供灵活性［H3］。由于使用了 8×8 整数变换、它的编码效率在保真度范围扩展（FRExt，Fidelity Range Extension）中得以进一步提高，而且对更复杂的视觉内容也能很好地工作。JPEG［JP1］是第一个静止图像压缩标准，它使用基于 8×8 块的 DCT 分解，而 JPEG2000 是一个基于小波的压缩标准，它提高了 JPEG 的编码性能，同时具有附加的特征如可扩展性和无损编码能力，而且对于平滑的空间数据具有最好的性能。JPEG 在低复杂度应用中性能良好，而 JPEG2000 在高复杂度较低比特率应用中表现很好。JPEG2000 具有超过 JPEG 率失真性能的优势。微软的 HD photo［JX4］是一个面向单色调摄影图像新的静止图像压缩算法，它保持着最高的图像质量或者最优的性能。JPEG-XR［JX3］（扩展范围），是 HD-photo 的标准，它高动态范围图像编码和性能成为吸引人的特征。它的性能接近于 JPEG2000，而计算和存储需求接近 JPEG。HD photo 给出的有损压缩图像，文

㊀　原文：I-Frame Only Coding——译者注

件大小是 JPEG 的一半但视觉质量好于 JPEG，而且无损压缩图像比原始图像小 2.5 倍。JPEG-LS［JL4］（无损的）是一个面向静止图像无损编码的 ISO/ITU-T 标准。此外，它也提供对于“近无损”压缩的支持。JPEG-LS 的主要目的是给出一个低复杂度解决方案，以最大可能的压缩效率用于无损图像编码。JPEG 使用霍夫曼编码，H.264/AVC 和 AIC 系统采用 CABAC 编码技术，HD photo 使用可逆的整数-整数-映射重叠双正交变换［J22］。LOCO-I（Low Complexity Lossless Compression for Images，图像的低复杂度无损压缩），是一种 JPEG-LS 的算法，它运用自适应预测、上下文建模和哥伦布编码。它允许有一个固定的最大样值误差，从而支持近无损压缩。

尽管以上提到的压缩技术是面向不同信号开发的，但它们针对静止图像压缩的压缩工作良好，因而具有比较价值。不同的软件如 AIC 参考软件、H.264 的 JM 软件［H30］、JPEG 的参考软件［JP10］、HD-photo 参考软件［JX4］、JPEG2000 的 JasPer［J1］、JPEG-LS 参考软件，用于不同编译码器之间的比较。我们使用比特率、不同的质量评价指标如 PSNR、SSIM 和复杂度进行评价。

下面的这些主题将在本附录中讨论。因为要实现 AIC 故对它进行了详细描述，进行比较的其他编译码器则简要说明。本附录讨论了用在软件中的不同设置和评价方法学。通过评价不同的测试图像获得结果，还包含了使用 AIC 参考软件不同大小的测试图像。

F.2　高级图像编码

高级图像编码（AIC）是一个静止图像压缩系统，它综合了 H.264 和 JPEG 标准算法，如图 F.1 所示，为了以较低的复杂度实现最好的压缩能力（以质量因数表述）。通过解决瑕疵问题和提高图像质量 AIC 可以开发。AIC 的性能接近 JPEG2000，且不 JPEG 好很多。AIC 运用帧内块预测，它最初用在 H.264 中来降低编码原始输入的比特数。AIC 和 H.264 都用到 CABAC 编码，而 AIC 用系数矩阵的位置作为上下文［AC1］。

可以观察到，AIC 的每个块都被修改以得到可能的最好压缩效率。

1. 概述

从 RGB 到 YCbCr 的颜色转换如图 F.2 所示，该转换使各通道更好的压缩成为可能，因为色度通道只有较少的信息内容。然后每个通道分成 8×8 的块用于预测。预测基于图 F.4 中所示的 9 种模式，这些模式基于以前编码和解码的块。色度通道使用相同的预测模式作为亮度中的对应块。将 DCT 应用于残差块时进一步降低了熵值。CABAC 用于编码比特率，它运用了一个上下文，其中常用的预测模式和 DCT 系数比很少使用的预测模式和系数耗用较少的比特数［AC1］。可以观察到，AIC 的每一块都被修改以获得可能的最好的压缩效率。

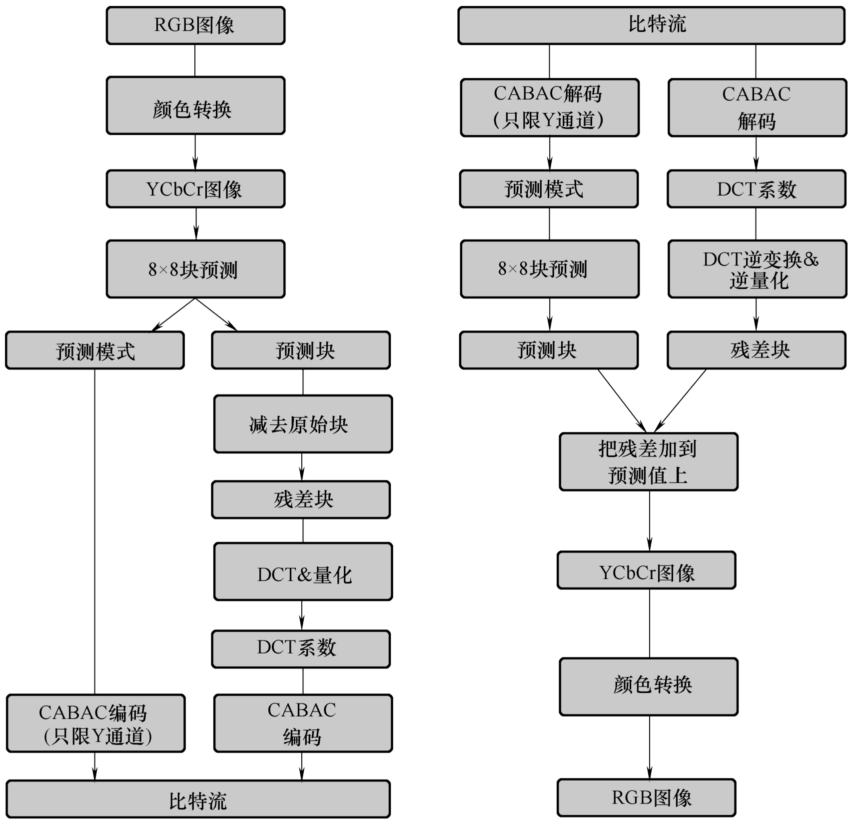


图 F.1　AIC 编码器和解码器的过程流［AC1］

2. 颜色转换

从 RGB 到 YCbCr 的颜色转换如图 F.2 所示，该转换使各通道更好的压缩成为可能，因为色度通道只有较少的信息内容。

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 0.257 & 0.504 & 0.098 \\ -0.148 & -0.291 & 0.439 \\ 0.439 & -0.368 & -0.071 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} + \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix}$$

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1.164 & 0 & 1.596 \\ 1.164 & -0.392 & -0.813 \\ 1.164 & 2.017 & 0 \end{bmatrix} \begin{bmatrix} Y-16 \\ Cb-128 \\ Cr-128 \end{bmatrix}$$

图 F.2　RGB 和 YCbCr 的颜色转换矩阵（针对 8 bit/像素的情况）

AIC 没有使用亚采样就实现了较高的质量/压缩比，而 H.264 和 JPEG 采样了亚采样。通过使用块预测和二进制算术编码这是有可能实现的。AIC 使用 4：4：4 格式，如图 F.3 所示。亚采样对图像质量有负面影响。

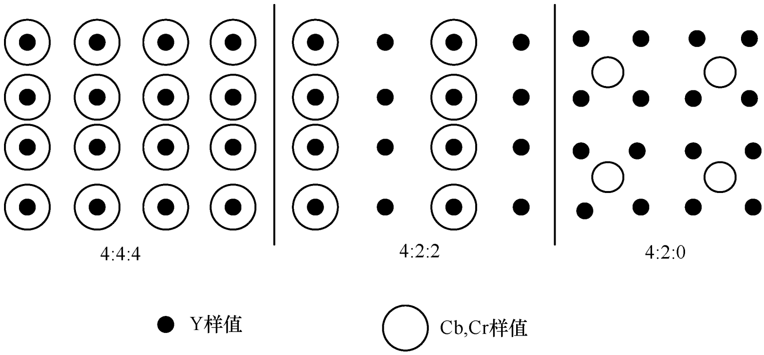


图 F.3　YCbCr 采样格式——4∶4∶4，4∶2∶2 和 4∶2∶0［B12］

3. 块预测

每个通道分为 8×8 的块以便预测。每个 8×8 块用从左到右自上而下的扫描行顺序进行编码。H.264 支持 4×4，8×8，和 16×16 的块预测算法，而 AIC 使用 4×4 的块算法，它可被推广到 8×8 的块情况。通过使用以前编码并解码的块进行预测。H.264 和 AIC 使用了 9 中预测模式来预测当前块，如图 F.4 所示。给出原始块和预测块之间最小差值的模式会被选中。预测需要所有像素的信息。第一块不能由以前的块预测得到。所以，用直流模式满足这一目的。为了降低复杂度，Y 使用的相同的预测模式被用于 Cb 和 Cr。从原始块中减去预测块可得到残差块。

4. AIC——块预测实现细节

用于块预测的不同模式如图 F.4 所示

5. DCT 和量化

DCT 作用于每个 8×8 的残差块。DCT 具有能量会聚的特性。使用了实际上没有丢弃比特的均匀量化。质量水平设置只是设置量化参数。AIC 采用浮点算法以产生最好质量的图像。

在 JPEG 中，DCT 系数以 Z 型顺序传输系数如图 F.5a 所示，而不是 AIC 所用的如图 F.5b 所示的扫描线顺序。Z 型扫描需要重排系数来构成零游程，它可用游程编码方法进行编码。CABAC 不需要重排系数。故游程编码是不必要的。

CABAC：

以上过程生成的预测模式和 DCT 系数必须存储在一个流中。AIC 使用 CABAC 算法最小化比特流。CABAC 利用不同的上下文编码符号。

算术编码能够编码分数比特数且性能超越霍夫曼编码，但较为复杂和缓慢。系数在矩阵中的位置可以成为上下文。这是可以获得的，因为 DCT 在高频区域中有高概率的零系数。AIC 所用的不同上下文有预测-预测模式、预测模式、系数图、最后的系数、大于 1 的系数、系数绝对值、编码块［AC1］。

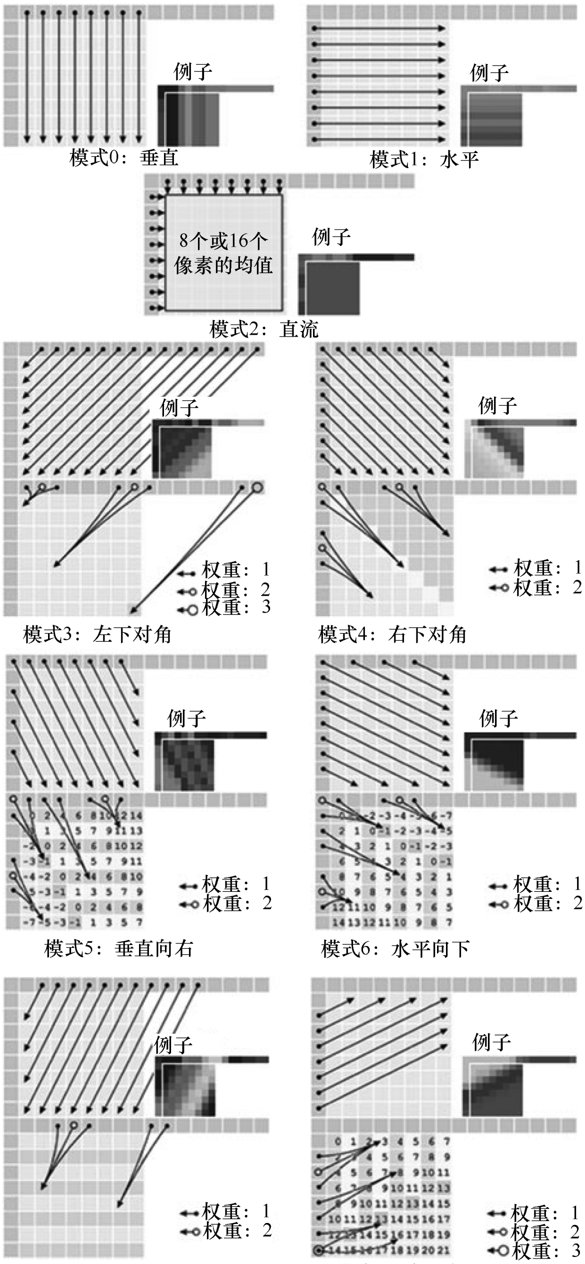


图 F.4　AIC 中用于预测的不同预测模式［AC1］

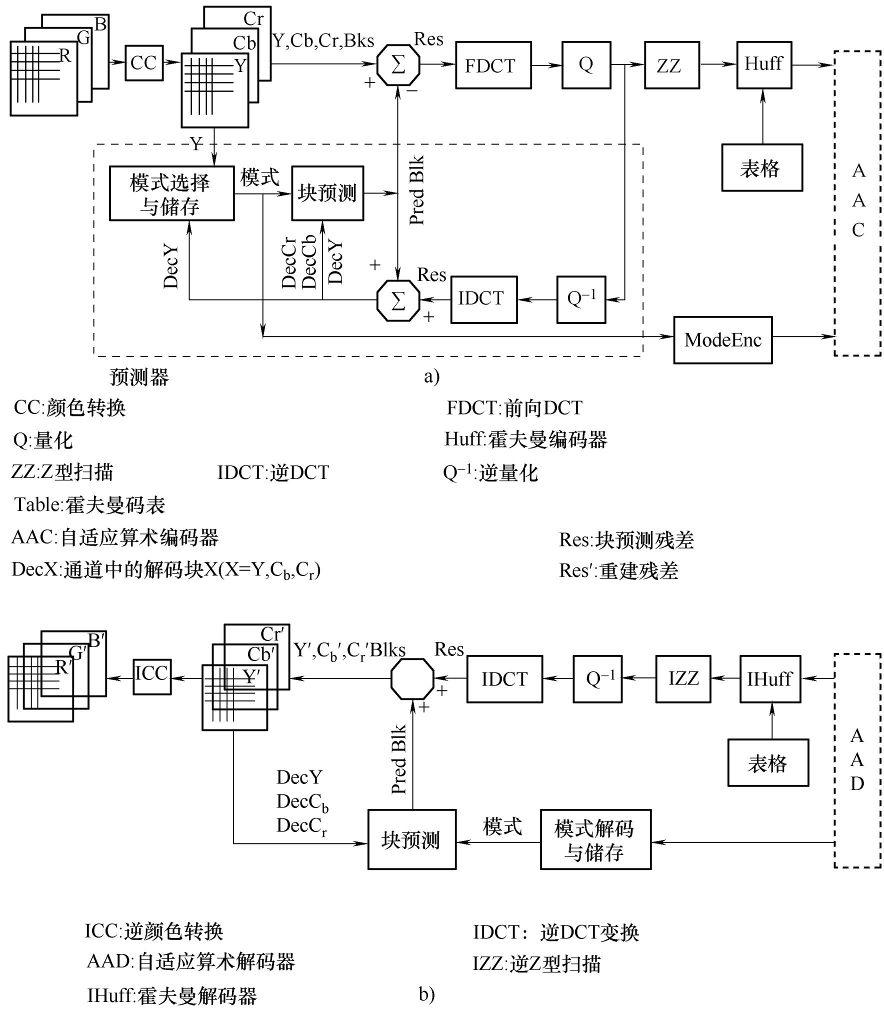


图 F.5　a）M-AIC 编码器　b）M-AIC 解码器

F.3　改进的 AIC

1. 编码器

如图 F.5a 所示，M-AIC 基于 JPEG 结构，加入了一个预测器。该预测器由五个部分组成包括 IDCT、Q^{-1}、模式选择与存储、块预测和一个加法器。预测器的功能是预测当前块，它是用以前上方行和左侧列的解码块进行编码的。原始图像的 R（红）、G（绿）和 B（蓝）分量不经过颜色转换模块的下采样转换为三个通道

Y（亮度）、Cb 和 Cr（色度）。也就是说，YCbCr 通道的格式是 4：4：4。AIC 使用与 JPEG 参考软件［JP10］相同的颜色转换方法。

三个通道 Y、Cb 和 Cr 依次压缩，Y 是第一个 Cr 是最后一个。与 JPEG 相同，每个通道分成互补重叠的 8×8 块，然后从左到右自上而下按扫描顺序逐块编码。编码每个 Y 通道的块时，要做的第一件事情是，通过在［AC1］预定的 9 个模式中全搜索选择块预测模式，它使绝对误差和（SAD，Sum of Absolute Difference）度量的预测误差最小化。9 种块预测模式编号为 0～8，叫作模式 0～模式 8，分别表示垂直、水平、直流、左下对角、右下对角、垂直向右、水平向下、垂直向左和水平向上。选出的预测模式存储下来然后不仅用于预测 Y 的当前块，还用于预测 Cb 和 Cr 的对应块，这些对应块与 Y 中的当前块具有相同的块索引号。编码的块预测残差（Res）利用快速浮点 DCT 算法变换到 DCT 系数。然后，DCT 系数被均匀地标量量化。相同的量化参数（QP，Quantization Parameter）用来量化整幅图像所有的 DCT 系数，包括所有的通道 Y、Cb 和 Cr。64 个 DCT 量化系数的二维矩阵通过 Z 型扫描顺序转化成一个一维序列。所有包括直流系数和交流系数的 64 个系数用与 JPEG 标准编码交流系数相同的算法进行编码。提出的算法中所用的编码所有通道 Y、Cb 和 Cr 的霍夫曼码表，与基本 JPEG 推荐的色度交流系数霍夫曼码表相同［JP6，JP10］。选出的预测模式由具有变长码算法的 ModeEnc 模块编码。如果当前块的预测模式与前一块相同，那么只输出 1bit 的 0，否则输出 1bit 的 1 后接 3bit 的模式编号信息，该信息是模式索引号本身或者是索引号减 1 如果当前块的索引号小于或大于前一块索引号。为组成一个压缩流，用 11 个字节来构造一个流首部，其中包括流格式标志、算法版本、量化参数、图像宽度、图像高度、原始图像的像素比特数、以及压缩模式的代码长度。压缩码流依次包括首部、预测模式代码、Y 通道残差（Y-Res）霍夫曼码、Cb-Res 霍夫曼码和 Cr-Res 霍夫曼码。我们编码器的末尾加入了自适应算术编码器（AAC：adaptive arithmetic coder）。选自［DC3］的自适应编码器源代码可以从［DC4］中下载。AAC 的输入 8bit 的符号，这些符号从压缩流（首部，预测模式代码，Y-Res、Cb-Res 和 Cr 残差的霍夫曼码）中逐字节抽取出来。AAC 的输出作为最终的压缩结果存储起来。

2. 解码器

提出的 M-AIC 解码器如图 F.5b 所示。事实上，解码过程是编码过程的倒置。来自编码器的比特流输入到 AAD（Adaptive Arithmetic Decoder，自适应算术解码器），产生了流首部、预测模式代码和 Y-Res、Cb-Res、Cr-Res 的霍夫曼码。预测模式码解码为预测模式并有 ModeEec 模块存储。通过类似于基本 JPEG 解码器［JP10］的解码算法得到当前块的残差。当前块的预测按照其预测模式从以前解码块中产生。重建残差加到预测值上生成重建的当前块。然后，Y、Cb 和 Cr 的所有块得以重建，三个通道 Y、Cb 和 Cr 由颜色逆转换模块转换到 R、G 和 B，参见图 F.2。

F. 4　H. 264 标准

与 MPEG-2 相比，H. 264 或 MPEG-4 第 10 部分目的是在相同的质量上以大约一半的比特率编码视频序列。它的目的还在于，以 CABAC 熵编码器、误差的稳健性和网络的友好性实现编码效率的明显改善。参数集概念、任意分片排序、灵活的宏块结构、冗余帧、切换预测帧和切换帧内编码帧促成了这个标准的抗误码/稳健性。自适应（方向）帧内预测（见图 F. 4）是促成该标准高效编码的因素之一［H3］。

每个档次规定了整个语法和限制比特流的一个子集，所有遵守这个档次的解码器都应该支持这个集合。在第一版中有三个档次：基本档次、主档次、和扩展档次。主档次面向数字存储媒体和电视广播而设计。H. 264 主档次作为高档次的子集，其设计的主要目的是压缩编码效率。保真度范围扩展［H11］在压缩效率方面给出了一个重大突破。该档次如图 F. 6 所示。

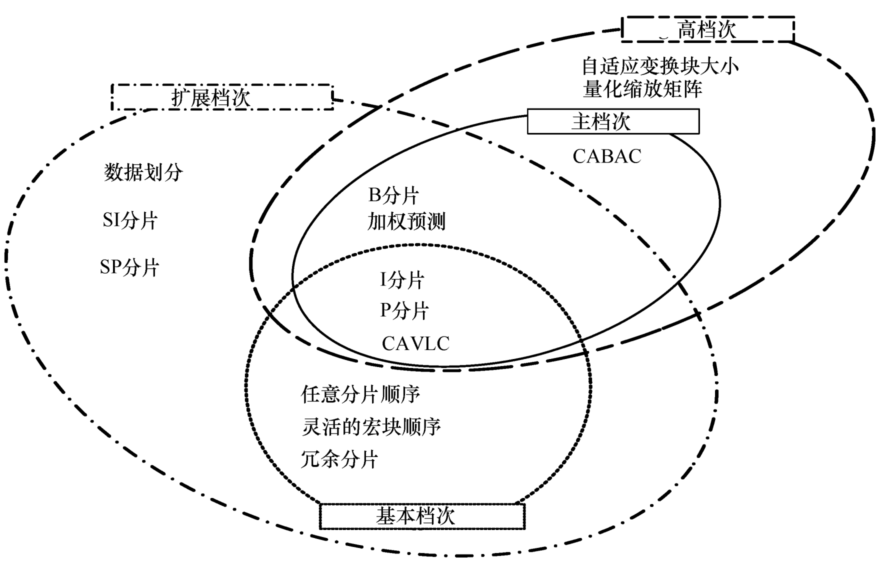


图 F. 6　H. 264 档次中的特定编码部分［H3］（2003 IEEE）

在保真度范围扩展中定义有四个高档次：高档次、高 10 档次、高 4：2：2 档次和高 4：4：4 档次。高档次支持面向高分辨率应用的 4：2：0 采样的 8 比特视频。高 10 档次支持 4：2：0 采样，且每个样值的表示精度高达 10 比特。高 4：2：2 档次支持高达 10bit 每样值。高 4：4：4 档次支持高达 4：4：4 色度采样且每个样值 12bit 的精度，从而支持高效的无损区域编码［H3］。

F. 4. 1　H. 264/AVC 主档次帧内编码

H. 264/AVC 主档次帧内编码和 JPEG2000 之间的主要差别在于变换阶段。这一阶段的特性也决定了量化和熵编码阶段。H. 264 使用基于块的编码，如图 F. 7 所示，就像帧间编码框架所用的块平移模型一样［J22］。采用了 4 ×4 大小的变换块而不是 8 ×8。H. 264 利用宏块的帧内预测来开发空间冗余，预测利用同一帧的相邻像素，这样就利用了块间的空间预测。

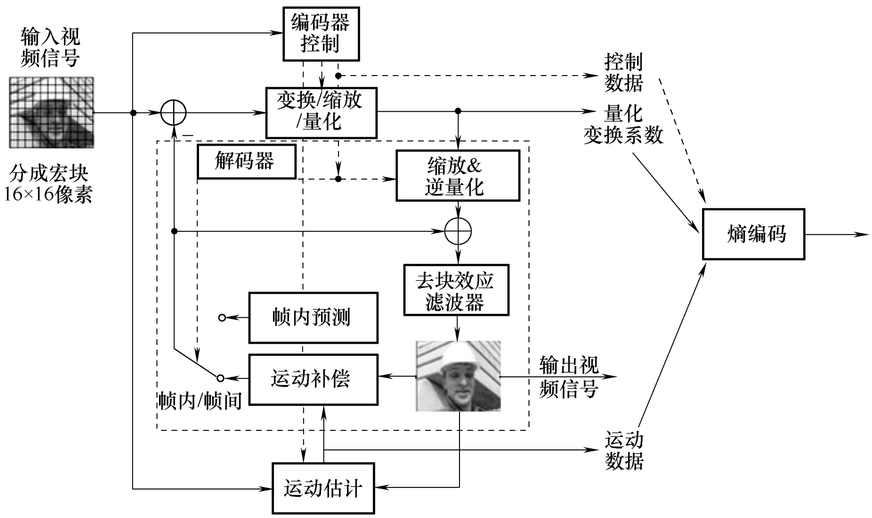


图 F. 7　H. 264/AVC 中宏块的基本编码结构［H3］（2003 IEEE）

使用空间预测以及 2 级小波变换的结果在平滑图像区域中是有效的。这一特征使得 H. 264 在高分辨率高质量应用中与 JPEG2000 可以相比拟。JPEG 即使运用了基于 DCT 的块编码也不能具有这样的可比性。如果利用上下文自适应的熵编码合理地考虑了相邻像素间的相关性，那么 DCT 编码框架与小波变换编码具有可比性。

在 H. 264 中，实施变换之后，系数进行标量量化、Z 型扫描并由 CABAC 进行熵编码。另一种熵编码 CABAC 通过在不同的 VLC 码表之间进行切换进行工作，这些码表的设计用到基于局部上下文的指数哥伦布码，上下文从所用的相邻块中收集获得，同时要牺牲一点编码效率［H3］。

F. 4. 2　H. 264/AVC FRExt 高档次帧内编码

改善 FRExt 编码效率的主要特征是 8 ×8 整数变换和所有的编码方法，以及与 4 ×4 和 8 ×8 整数变换间自适应选择相关的预测模式。其他特征包括［H11，J22］：

- 较高的颜色表示分辨率如 YUV 4：2：2 和 YUV 4：4：4，参见图 F.2；
- 在甚高分辨率和高的比特率下，8×8 块的加法是一个关键因素；
- 实现很高的保真度——甚至对于选择性的无损视频表示。

基于内容的自适应二进制算术编码（CABAC，Context-based Adaptive Binary Arithmetic Coding）：

为了得到好的压缩，CABAC 利用了算术编码。CABAC 编码过程如图 F.8 所示，包括三个基本步骤。

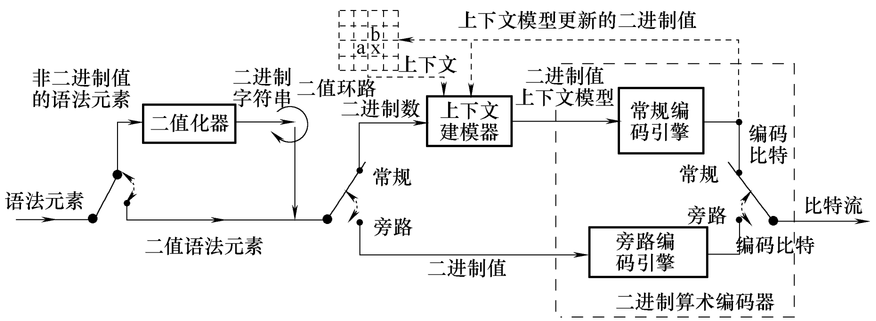


图 F.8　CABAC 的框图［H25］（2006 IEEE）

步骤 1：二值化——在输入给算术编码器之前，把非二进制符号映射到二进制序列中。

步骤 2：上下文建模——它是一个概率模型，基于以前编码的语法元素定义了一个或多个元素。

步骤 3：二进制算术编码——基于所选的概率模型对语法元素进行编码。

F.5　JPEG

JPEG 是第一个面向连续色调静止图像的 ISO/ITU-T 标准［JP1］。它允许对静止图像进行有损和无损的编码。JPEG 以最低的复杂度给出了有损压缩良好的压缩结果。JPEG 有几个定义的模式包括基本、渐进和分级模式。基本模式支持只支持有损压缩，是相当流行的。在基于 DCT 块压缩的帮助下，可以实现平均 15:1 的压缩比。利用预测编码压缩技术可以实现无损编码，它包括差分编码、游程编码和霍夫曼编码。JPEG 进行按照 HVS 加权的均匀量化。量化系数要进行 Z 型扫描（见图 F.9a），因为它允许以从低频到高频分量的顺序进行熵编码［JP1］。

JPEG 基本（有损）算法过程流图如图 F.10 所示。

基本模式的过程流，编码过程开始于彩色图像的压缩转换，紧接着是基于 8×8 块的 DCT（对于灰度图像过程流从这里开始），量化，Z 型扫描，用霍夫曼码表

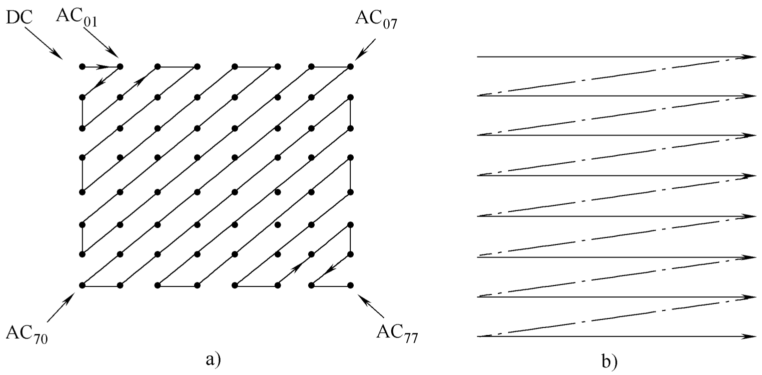


图 F. 9　a）Z 型扫描［JP1］　b）扫描行顺序

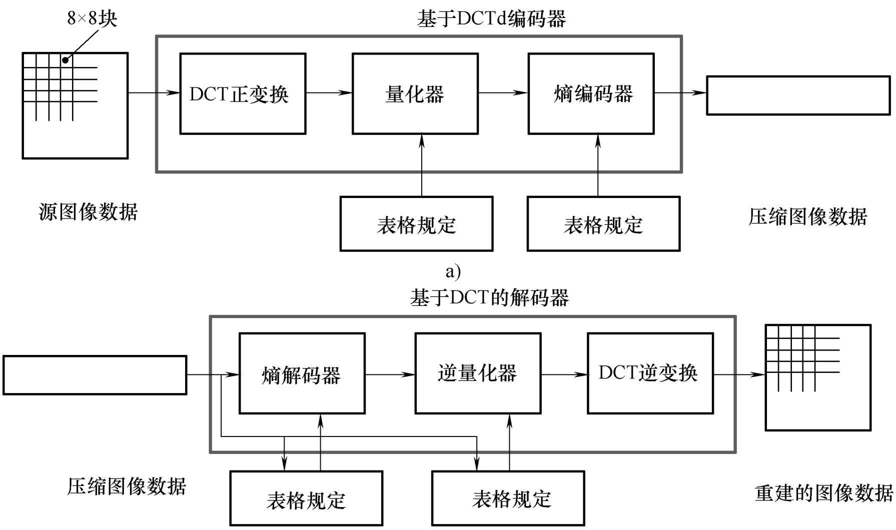


图 F. 10　a）JPEG 编码器的框图　b）JPEG 解码器的框图［JP1］

进行熵编码；对于解码过程，反之亦然。

不同的量化矩阵用于亮度和色调分量。每 64 个 DCT 系数的量化步长规定在一个量化表里，它对于所有的块都是相同的。所有块的直流系数使用预测方案单独编码。

质量因数“Q”用量化表来设置，我们可观察到在各种各样的范围中不同种类的瑕疵［JP1］。

F. 6　JPEG2000

JPEG2000 [J8] 是一个图像压缩标准，支持灰度或彩色图像的有损和无损压缩。除了压缩能力之外，JPEG2000 还支持卓越的低比特率性能（没有牺牲高比特率时的性能）和感兴趣区域编码，随机访问规定图像区域的 EBCOT（Embedded Block Coding with Optimized Truncation，优化截取嵌入式块编码）、以及差错恢复。它还支持灵活的文件格式和图像的渐进解码，允许保真度和分辨率从无损到有损都会出现。它的结构使它适合于广阔的应用，从便携式数码相机到高端印前设备、医学成像和其他重要领域。

JPEG2000 基于 DWT（离散小波变换），标量量化，上下文建模、算术编码和后压缩率分配。小波变换具有超越基于 DCT 压缩 3 分贝的改善 [J2]。无损压缩是变换、熵编码的结果。熵编码在块内完成，通常是在每个子带内 64 × 64 的大小。DEW 可用提供无损编码的可逆滤波器来完成；或者用提供较高编码效率但不可能无损的不可逆滤波器来完成。算法用‘后压缩率分配’将编码数据组织起来形成所谓的层，也就是质量级别，然后输出到数据包的码流中。JPEG 在统一的算法中提供了分辨率、SNR 和位置的渐进性，或者它们的任意组合、可解析的码流、差错恢复、任意形状的感兴趣区域、随机访问（子带块级别）、有损和无损编码等。

JPEG2000 编码过程如下 [J11]：

- 将图像分解成分量；
- 图像及其分量再被分解成矩形的片，片分量是原始图像或重建图像的基本单元；
- 小波变换作用于每一片，把它分解成不同的级别；
- 这些分解级别由系数子带组成，这些子带描述了片分量局部区域的而不是整个片分量的频率特性；
- 系数子带被量化再分割成“码块”构成的矩形阵列；
- 对“码块”系数的比特面进行熵编码；
- 可以使得某个 ROI 以高于背景的质量编码；
- 在比特流中加入标识使得差错恢复成为可能；
- 码流的前端有一个主首部，它描述了原始图像和各种分解以及编码风格，它们得用于定位、抽取、解码和重建图像，使图像具有期望的分辨率、保真度、感兴趣区域或其他特性（见图 F. 11 和图 F. 12）。

JPEG2000 在很多领域如数字影院、视频监控、军事和医学成像、计算机图形学或动画制作获得了成功的应用，这归功于一些重要特征的组合发挥，如可扩展性和同时具有的高位深。

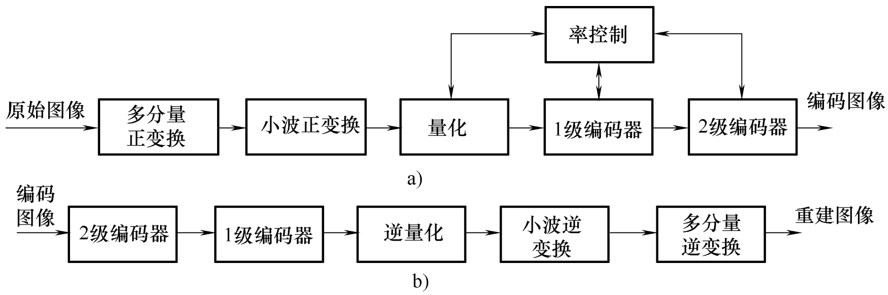


图 F. 11　JPEG2000 编译码器结构［J5］(2000　IEEE)
a) 编码器结构　b) 解码器结构

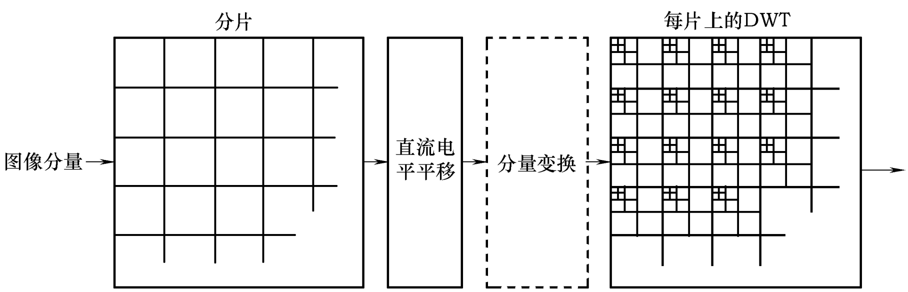


图 F. 12　分片，直流电平平移、压缩变换、每个图像分量的 DWT［J10］(2001 IEEE)

F. 7　JPEG XR

JPEG XR［JX3］是一种编码的文件格式，其设计明确面向下一代数码相机和连续色调摄影内容的存储，主要基于微软的技术，也叫作 HD photo。它支持很多颜色格式，包括定点和浮点数值表示的 n 通道编码、位深变化，为各种数据压缩环境提供了一种方法。其终极目标是支持宽范围的颜色编码，保持与现存格式的前向兼容形以及维持设备实现的简单化。它还提供同样的无损算法和有损压缩。

HD photo 格式［JX4］是一种用 JPEG-XR 标准化的新文件格式。就像 JPEG-2000 一样，微软的 HD photo 的工作基于很多除压缩能力之外的高级特征，如有损/无损压缩，比特率可扩展性，编辑，感兴趣区域解码，不用除法的整数实现等。HD 最小化了讨厌的空间瑕疵，保持了高频细节，且在这样方面性能优于其他有损压缩技术。

HD photo 是一个基于块的图像编码器，类似于传统的图像编码范例：颜色转换，变换，系数扫描，标量量化和熵编码。HD Photo 的主要模块包括变换阶段和系数编码阶段。HD photo 运用了一种可逆的整数-整数映射的重叠双正交变换

（LBT）作为它的去相关引擎。算法的可逆特性既支持有损又支持无损压缩。因而，它简化了系统的整体实现。HD photo 的编码器包含了许多自适应元素：自适应系数扫描，灵活的量化，块间系数预测，自适应 VLC 码表切换等等，如图 F.13 所示。JPEG XR 支持若干高级像素格式来避免不同无符号整数表示之间转换的限制和复杂性。

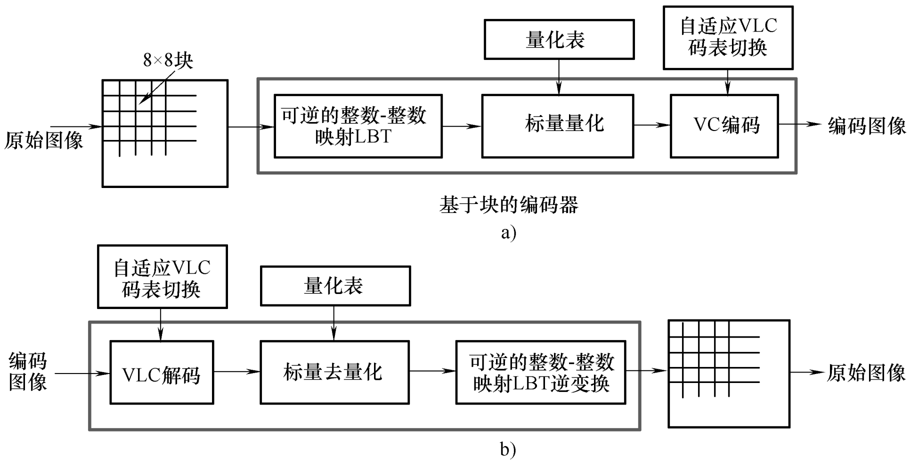


图 F.13　a）JPEG XR 编码器的框图（HD photo 编码器）
b）JPEG XR 解码器的框图（HD photo 解码器）

这一特征使得对图像数据数值编码可以使用灵活的方法，使得编码器和解码器的实现具有低的复杂度。

F.8　JPEG-LS

JPEG-LS［JL4］压缩算法是已制定的面向彩色和灰度图像无损压缩标准之一。惠普公司提出了一个更简单的低复杂度预测器［JL2］。LOCO-I（图像的低复杂度无损压缩）是一个面向连续色调图像的无损压缩算法，它综合了霍夫曼编码的简单性和上下文模型的压缩潜力。无损图像压缩方案通常包括两个不同且彼此独立的部分：建模和编码。建模部分可以描述成一个归纳推理问题，按照某种预定顺序（如光栅顺序）逐像素观察一幅图像。为了性能高效，结合一个自适应选择的哥伦布码的扩展族系和一个面向低熵图像区域编码的嵌入式字母扩展集，去调整这个模型。LOCO-I 算法获得了与基于算术编码最先进方案所得到的类似的压缩比。

JPEG-LS 可用于无损和近无损模式来压缩图像，这两种模式可被定义为：

- 无损压缩：顾名思义，不会有信息损失，而且图像可以重建得完全和原始图像一样。

- 近无损模式：重建图像分量的每个样值与原始图像的对应值相差至多一个预定的（小）量“NEAR”

在 JPEG-LS 压缩中有三种模式。它们是：

1）逐平面模式——图像分量一个接着另一个进行压缩。只有灰度图像以此模式进行压缩。

2）行交织模式——在交织模式中，每次处理每个图像分量的一行，灰度和彩色图像都可以此模式进行压缩。

3）样值交织模式——每次处理每个图像分量的一个样值。只有彩色图像以此模式压缩。

在此附录中，JPEG-LS 代码已被优化，而且只在行交织模式下进行测试，因为它既支持灰度图像又支持彩色图像。

JPEG-LS 方法提供下列优势：

- 算法设计的低复杂度；
- 实现的低成本；
- 在质量、压缩效率和处理速度方面更接近与无损压缩。

F.9 JPEG-LS 算法

好的预测和有效的残差编码足以对于大多数自然图像源实现具有竞争力的压缩性能。在 JPEG-LS 算法中，将非线性算法用于预测。上下文模型和哥伦布编码器被一起用于有效地压缩预测残差 ，如图 F.14 所示。

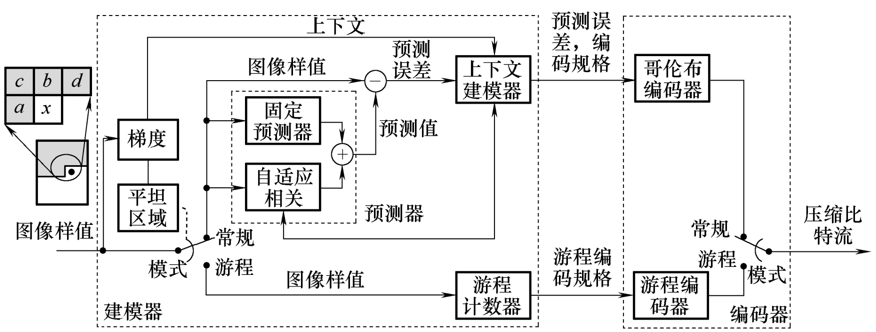


图 F.14 JPEG-LS 框图［JL2，JL4］（2000 IEEE）

LOCO-I 的描述

LOCO-I 中的预测与建模单元基于图 F.15 所描述的因果模板。

1. 预测

预测方法是中值自适应预测的一种变体，其中的预测值是 a、b、c 像素的中值。起始预测用下述算法得到。

```
if (c >= max (a, b))
      x = max (a, b);
else
{
      if (c <= min (a, b))
         x = min (a, b);
      else
         x = a + b - c;
}
```

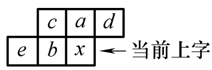


图 F. 15　LOCO-I 的一种因果模板［JL3］（1996 IEEE）

然后，用那个特定上下文中预测误差的平均值来细化起始预测值［JL2］。

2. 上下文建模

上下文建模方案的关键目标是减少参数个数。

1）编码分布：连续色调图像预测误差的分布常常可以用拉普拉斯分布来近似，即中心为 0 两侧呈指数下降的曲线。对于每个上下文，编码器基于过去的性能自适应地在一个霍夫曼码的有限集中选择最好者以匹配指数下降的分布。因为这些分布假定以 0 为中心，一个参数（例如，对应上下文中误差幅度的均值）就足以描述每一分布的特征［JL2］。

2）确定上下文：JPEG-LS 的上下文也反映了像素值的局部变化。LOCO-I 中影响当前预测残差编码的上下文，从如下几个差值中建立起来 $g_1 = d - a$，$g_2 = a - c$，$g_3 = c - b$ 和 $g_4 = b - e$，如图 F. 15 所示。上下文从以前编码出现的预测误差中建立。因为需要进一步减少参数，每个差值 g_j，$j = 1, 2, 3, 4$ 被量化到少量的或近似等概的区域中（对于 $j = 1, 2, 3, 4$ 是相同的区域）［JL2］。

3. 编码

LOCO-I 组合了霍夫曼编码的简单性（与算术编码相比）和上下文模型的压缩潜力［JL4］。预测误差使用自适应选择的基于哥伦布码的码进行编码，对于几何分布的序列而言它也是最优的［JL2］。

1）序贯参数估计：在基于上下文的方法中序贯方案是强制实施的，因为给定上下文中的编码像素在图像中未必是相邻的，因而不易被阻挡。

2）取消偏移：哥伦布-莱斯码［B12］很大程度上依赖于预测误差的分布，它以 0 为中心，两侧呈现对称的指数下降趋势。尽管这些假定在无记忆模型中通常是满足的，但在基于上下文的模型情况下，情形是相当不同的，其中预测残差的规则的上下文相关的偏移并非少见。这些规则偏移可以在哥伦布-莱斯编码器的编码性

能中产生明显的恶化现象。为了减轻规则偏移的影响，LOCO-I 使用了误差反馈以把预测残差的分布居中［JL2］。

4. 嵌入式字母扩展集

LOCO-I 通过把字母扩展集嵌入到上下文训练过程中解决了由于歪斜分布造成的霍夫曼码或哥伦布-莱斯码的冗余问题。

因为具有多种模式发挥作用的能力，它在可能包含图像和文本的复合文档中性能优异［JL4］。在接近于更低数量级的复杂度上，LOCO-I 位于实现最好压缩比（由 CALIC 给出）的方案中很小的百分比之内。

它可被用于压缩卫星、医学、文档、绘画和计量生物学图像，或者是无损的或者是近无损的方式。在某些应用中，例如高速扫描和卫星图像传输中，需要接近实时地处理大量图像，硬件的解决方案非常可贵。在压缩能力方面，JPEG-LS 算法与其他主流的无损压缩算法可以相比拟，但比其他主流算法的复杂度更低。JPEG-LS 的另一个特征是，它允许近无损压缩。用户可以通过规定好允许的容差能提高压缩比和执行速度。

JPEG-LS 可被用于静止图像和视频应用中。它可以无损或近无损的方式用于压缩卫星、医学、文档、绘画和生物计量学图像。它对于成本敏感和不需要任何 JPEG2000 功能的嵌入式应用中如渐进比特流、差错恢复、感兴趣区域编码（ROI）等，工作良好。

F.10 主要差别

AIC，JPEG，JPEG2000，JPEG-LS 和 JPEG-XR 编译码器之间的主要差别在于变换阶段。JPEG2000 借助全局离散小波变换（DWT）或更一般的小波包分解；而 H.264 和 HD Photo 则选择基于块的编码框架，使用相同的 16×16 宏块大小和很像 DCT 的核心 4×4 块变换。JPEG 和 AIC 使用离散余弦变换（DCT）去除图像的相关性。H.264 和 HD Photo 变换阶段的主要不同在于这两个编码器处理块间去相关的方式。H.264 很大程度上依赖当前块从其相邻块的空间预测。而 HD Photo 在把块输入到像 DCT 的核心 4×4 块变换之前，利用一种重叠运算沿着块边界对像素执行预处理。JPEG 和 AIC 变换阶段之间的主要差别是去相关过程所作用的基于块的系数。在 JPEG 中去相关过程作用于原始图像的所有系数，而在 AIC 中，它作用于残差块的系数。等效地，重叠运算和核心块变换的组合产生了重叠变换。类似于 JPEG2000，HD Photo 的整个变换步骤用二值有理数提升步骤来重建，这样就把整数映射成整数，并具有完美的可逆性，形成一个统一的从无损到有损的编码框架。相反，H.264 和 AIC 从残差编码中实现无损的压缩。另一个明显的区别是，在熵编码阶段每个编码器调节其基于上下文的自适应模型来利用其变换系数和/或参数的特定行为。H.264/AVC 使用空域的帧内预测。AIC 遵循同样的技术。这避免了帧

间编码宏块运动补偿带来的误差传播。另一方面，所有以前的视频编码标准像 H. 263 和 MPEG-4 视觉都在变换域使用帧内预测［B9］。LOCO-I 的性能明显地超越了其他复杂度相当的一次遍历方案（例如 JPEG-霍夫曼），而且它达到了类似于或优于基于算术编码（JPEG-算术）更高复杂度的方案［JL4］。与 JPEG 和 JPEG-LS 相比，JPEG2000 的复杂度相对高一些。

F. 11 评价方法

F. 11. 1 图像测试序列

在用 AIC 的评价中，相同和不同分辨率的各种彩色和灰度图像用于全面地评价它们的性能。我们会考虑用来评价不同纹理和图案的测试图像，以便于能够充分仔细地分析图像，以及使这些图像与不同软件在文件格式方面等具有兼容性。

F. 11. 2 编译码器的设置

在编码实验中，向公众开放的软件实现方案被用于 AIC、H. 264/AVC、JPEG-基本模式、JPEG2000、HD photo 和 JPEG-LS。参考软件 JM18［H30］用作 H. 264/AVC 的编码器，测试序列的每一帧以 I 帧模式编码。对于 JPEG，使用 JPEG 基本模式的参考软件［JP10］。这个软件可以处理多种图像数据格式如 PGM/PPM，GIF，Windows BMP。对于 JPEG2000 编码，使用 M. D. Adams 的 “JasPer”(版本 1. 900. 1) 软件［J1］。这一软件能够处理许多图像数据的格式如 PGM/PPM，windows BMP，但并非接受所有的 BMP 文件。对于 JPEG2000 的情况，用软件编码每一帧以达到压缩因数下的目标比特率，对于多分量图像这有完善的定义。HD photo 参考软件［JX4］支持 BMP、TIF 和 HDR 格式。JPEG 和 HD photo 参考软件都用来编码每一帧一达到目标质量因数，从而间接地控制了有损编码的比特率。JPEG-LS 参考软件［JL1］由 HP 实验室提供，实现 JPEG-LS 的有损压缩。它只支持 PGM/PPM 图像格式作为编码器的输入，JLS 格式作为编码器的输出。

H. 264/AVC JM18 编码器［H30］的配置选择如下：

- ProfileIDC =77 # 档次识别码（77 = 主档次，FREXT Profiles：100 = 高档次）
- LevelIDC =40 # 级别识别码（例如 20 = level 2. 0）
- IntraProfile =1 # 为 FRExt 激活帧内档次（0：假，1：真）
- Deblocking filter：off
- QPISlice =12 # I 分片的量化参数（0-51）
- YUVFormat =1 # YUV 格式（0 =4：0：0，1 =4：2：0，2 =4：2：2，3 = 4：4：4）.

JM18 软件的命令行输入内容是：

编码器：lencod- f encoder. cfg

解码器：ldecod- i bitstream. 264- o output. yuv- r reference（input）. yuv.

JPEG 基本模式软件［JP10］的命令行输入内容如下：

输入图像可能是 bmp 或 ppm 格式。

编码器：cjpeg- quality N inputfile. bmp outputfile. jpg

其中的质量因数“quanlity—N”表示调节图像质量的缩放量化表。质量因数 quality 在 0（最差）到 100（最好）之间变化；默认值是 75。

解码器：djpeg- outfile outputfilename. bmp- outputfileformat inputfile. jpg

JPEG- 2000 软件［J1］的命令行输入内容如下：

编码器：jasper- input inputfilename. bmp- output outputfilename. jp2- outputformat jp2- O rate =0. 01

或者 jasper- f inputfilename. bmp- F outputfilename. jp2- T jp2- O rate =0. 01　其中 rate 规定了正实数形式的目标比特率。Rate =1 对应没有压缩。Rate 和比特每像素（bpp）由以下表达式关联：对于彩色图像压缩比 =24/bpp =1/rate，对于灰度图像 rate = bpp/8。

对于微软的 HD Photo［JX4］，所有选项都设置为默认值，唯一的控制来自质量因数设置：

- 没有分片；
- 变换阶段中的一级重叠；
- 没有颜色空间亚采样；
- 空间比特流顺序；
- 没有跳跃地包含所有子带。

WMEncApp 命令行把某种没有压缩的文件格式转换为等效的 HD photo 文件。例如，要利用合理的高质量有损压缩从典型的 24bit 的 . bmp 创建一个 HD photo 文件，命令行的内容是：wmpencapp- i input. bmp- ooutput. wdp- q 10。

在有损压缩中，增大质量因数“q”导致降低 PSNR。q = 0 是无损压缩的情况。

WMPDecApp 命令行把 HD photo 文件转换为不同的未压缩文件格式。例如：wmpdecapp- i input. wdp- o output. bmp- c 0，其中的“c”表示格式，对于 24bit/像素的 RGB 图像，c =0；对于 8bit/像素的灰度图像，c =2。

为了对比图像，可以使用下列命令。

Imagecomp ImageFileName1 ImageFileName2［- i InputFormat］［- M 0/1］

其中 M（模式）=0（默认）给出 MSE 和 PSNR。

JPEG- LS 软件［JL1］的编码器设置如下。解码器不必要改变默认值因为它们遵循编码器的设置。

- 图像应该是 ppm 或 pgm 格式；

- 在专题中考虑行交织模式；
- 误差值在 1 到 60 之间变化。误差值为 0 对应没有压缩；
- T1、T2、T3 是阈值。尽管给定了设置，下列条件仍然需要满足。误差值 + 1 < T1 < T2 < T3。
- 本专题考虑默认预设值为 64。

F. 11. 3　主观对客观图像质量测度

无损和有损压缩用不同的方法评价压缩质量。在无损情况下，评价压缩的标准有压缩比、运行时间等，任务比较简单；但在有损压缩中，是比较复杂的，因为应该评价重建图像引入的质量下降的类型和度量［J14］。

图像质量评价的目的是准确度量原始图像和重建图像之间的差异。客观质量测度如 PSNR，度量原始和重建图像单个像素之间的差别。JPEG 编码器产生较大的 MSE 和 PSNR。

SSIM［Q13］的设计是为了改善传统测度指标如 PSNR 和 MSE［已被证明与人类视觉感知系统（HVS）不一致］。SSIM 指标给出假定，即 HVS 很大程度上受到场景内结构信息的影响。SSIM 指标是一个全参考测度，换言之，图像质量的度量结果基于作为参考的原始未压缩或无失真图像。SSIM 度量系统如图 F. 16 所示。

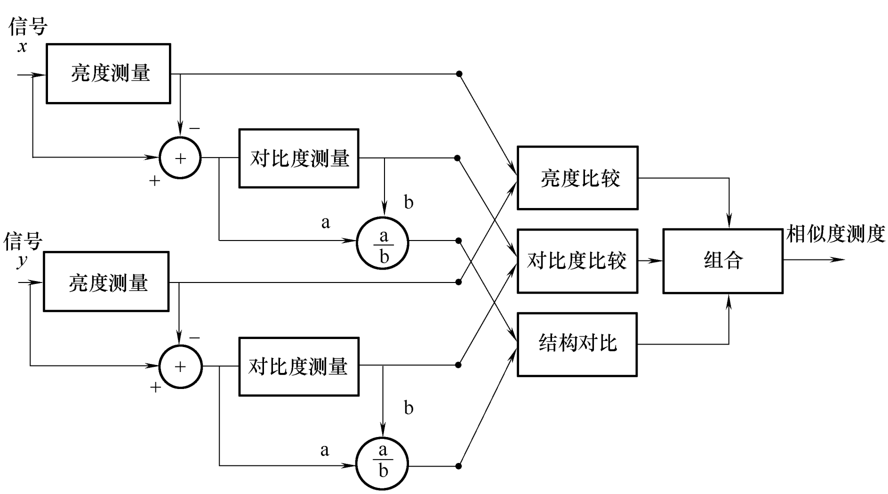


图 F. 16　结构相似度（SSIM）度量系统［Q13］（2004 IEEE）

SSIM 指标基于三个不同测度的评价，亮度、对比度和结构比较测度，分别计算如下［Q13］：

$$l(\boldsymbol{x},\boldsymbol{y}) = \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1}$$

$$c(\boldsymbol{x},\boldsymbol{y}) = \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2}$$

$$s(\boldsymbol{x},\boldsymbol{y}) = \frac{\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_3}$$

式中，$\boldsymbol{x}$ 和 $\boldsymbol{y}$ 对应我们要去匹配的两个不同信号，即两幅不同图像中的两个不同块；μ_x，σ_x^2和 σ_{xy}分别是 $\boldsymbol{x}$ 的均值、$\boldsymbol{x}$ 的方差和 $\boldsymbol{x}$ 与 $\boldsymbol{y}$ 的协方差；而 C_1、C_2和 C_3是常数由 $C_1=(K_1L)^2$，$C_2=(K_2L)^2$，和 $C_3=C_2/2$ 给出；L 是样值数据的动态范围，也就是说，对于 8bit 的像素 $L=255$；$K_1 \ll 1$ 和 $K_2 \ll 1$ 是两个标量常数。给定如上测度，结构相似度可以计算为

$$\mathrm{SSIM}(\boldsymbol{x},\boldsymbol{y}) = [l(\boldsymbol{x},\boldsymbol{y})]^{\alpha}[c(\boldsymbol{x},\boldsymbol{y})]^{\beta}[s(\boldsymbol{x},\boldsymbol{y})]^{\gamma}$$

式中，α，β 和 γ 定义了赋予每个测度的不同的重要性。

在另一方面，MS-SSIM 测度是 SSIM 的一个推广形式，它在不同的尺度上计算这些测度，然后用一个方程将它们组合起来，方程形式为［Q10］：

$$\mathrm{MSSSIM}(\boldsymbol{x},\boldsymbol{y}) = [l_M(\boldsymbol{x},\boldsymbol{y})]^{\alpha_M}\prod_{j=1}^{M}[c_j(\boldsymbol{x},\boldsymbol{y})]^{\beta_j}[s_j(\boldsymbol{x},\boldsymbol{y})]^{\gamma_j}$$

式中，M 对应我们已经用于计算的最大尺度；$j=1$ 对应图像的原始分辨率。SSIM 测度对于 YUV 和 RGB 颜色空间都可运行。

通常的瑕疵是：

- 块效应——由于基于块的 DCT 编码方案。

所以，我们可以在 AIC 和 JPEG 中观察到它。HD-photo 已经降低了块边界瑕疵。

- 抖动效应——来自基于小波的编码器。

JPEG2000 由于它的压缩算法而受到这种瑕疵的影响。

- 振铃——量化的结果。在亮度和色调分量中都恢复出现。

所有的编译码器都用到量化。所以，这是个重要因素。

- 颜色渗透——由于色度亚采样。

在 AIC，颜色渗透可被忽略因为它没有用到亚采样。所有其他的编译码器有这个瑕疵。

F.12　结论和未来的工作

本专题实现了如图 F.1 所示的 AIC 编码器和译码器，并用 AIC 参考软件对比了结果。据发现，M-AIC 的结果类似于 AIC。因而，没考虑用熵编码的 CABAC 通过降低复杂度获得了一个相对比较好的结果。这些结果与其他压缩技术就比特率、客观测度（质量—PSNR，SSIM）进行了比较，用到的实现这些技术的软件是 H.264 的 JM 软件［H30］、JPEG 基本模式参考软件［JP10］、JPEG2000 的 JasPer

[J1]、HD-photo 参考软件 [JX4] 和 JPEG-LS 参考软件 [JL1]。从结果中观察发现，AIC 与其他复杂度相当高的编译码器可相比拟。基于 SSIM 测量指标，AIC 性能超越高达5.5bit/像素，而且甚至超出了这个 bit/像素范围仍然保持着竞争力。由于这样优秀性能，在很广泛的领域如摄像机市场、互联网浏览、多媒体产品诸如移动电话和娱乐应用中都可以使用 AIC。不同的测试图像用于评价各种图像纹理和图案，以便充分详细地研究图像压缩技术。

这个专题还可被推广去比较无损压缩。CABAC 的较早实现方案可以作为未来的一个研究内容。

原始的和输出解码的图像：

Lena（512×512×24）


原始图像


AIC质量因数-5,2.37bit/像素,
36.61dB,SSIM-0.914


HDphoto质量因数-28,2.88bit/像素,
37.74dB,SSIM-0.928


JPEG-LS误差值-11,28bit/像素,
32.425dB,SSIM-0.818


JPEG2000比特率=0.12,
2.95bit/像素,37.53dB,
SSIM-0.923


H.264量化参数-16,2.83bit/像素,
46.81dB,SSIM-0.917


JPEG基本模式质量因数-94,
2.94bit/像素,35.6dB,
SSIM-0.926

SSIM 结果：Lena（512 ×512 ×24）——AIC

原始图像 [重建图像，MSE =14.2014 平均 SSIM=0.91476] [SSIM图]



JPEG

原始图像 [重建图像，MSE =17.9004 平均 SSIM=0.90267] [SSIM图]



JPEG 2000

原始图像 [重建图像，MSE =11.484 平均 SSIM=0.92336] [SSIM图]



JPEG - XR(HD Photo)

原始图像 [重建图像，MSE=10.9338 平均 SSIM=0.92815] [SSIM图]



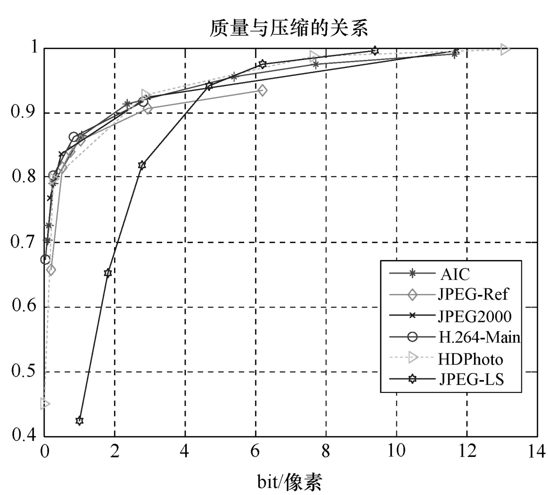
JPEG-LS



H.264

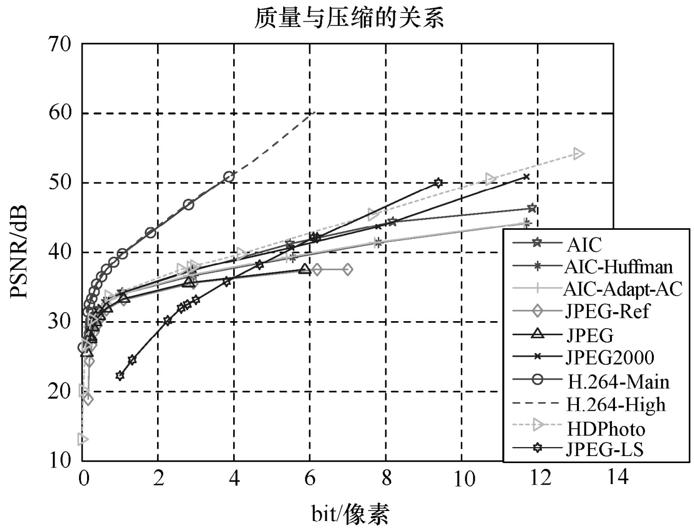


以 SSIM 作为失真测度对 Lena（512 × 512 × 24）的仿真结果



仿真结果和原始测试图像

Lena（512 ×512 ×24）

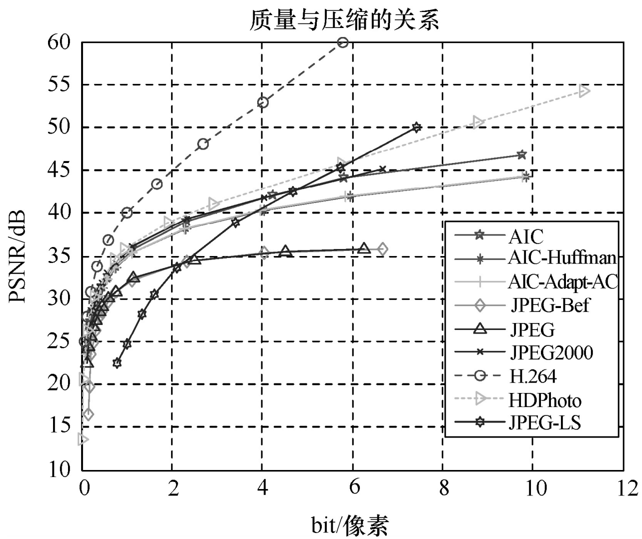




AIC、JPEG、JPEG 2000、HD Photo、H.264 和 JPEG LS 的仿真结果

M－AIC			JPEG2000			HD photo			H.264			JPEG-baseline			JPEG-LS		
比特率/(bit/像素)	PSNR/dB	压缩比(CR)	比特率/(bit/像素)	PSNR/dB	压缩比(CR)	比特率/(bit/像素)	PSNR/dB	压缩比(CR)	比特率/(bit/像素)	PSNR/dB	压缩比(CR)	比特率/(bit/像素)	PSNR/dB	压缩比(CR)	比特率/(bit/像素)	PSNR/dB	压缩比(CR)
0.15253	26.91	157.346096	0.17	28.74	141.176471	0.023621	13.04	1016.045	0.06	26.24	400	0.17139	18.92	140.031507	1.0155	22.17	23.633678
0.17636	28.25	136.08528	0.22	29.48	109.090909	0.033966	20.04	706.58894	0.17	31.3	141.17647	0.21921	24.29	109.484056	1.3393	24.56	17.9198089
0.21967	29.15	109.254791	0.27	30.24	88.8888889	0.10373	26.62	231.3699	0.21	32.31	114.28571	0.26511	26.64	90.5284599	2.2652	30.136	10.5950909
0.2988	30.16	80.3212851	0.34	31.04	70.5882353	0.26089	30.28	91.992794	0.26	33.27	92.307692	0.33878	28.49	70.8424346	2.641	31.797	9.08746687
0.3613	30.86	66.4267921	0.39	31.46	61.5384615	0.73004	33.51	32.874911	0.33	34.38	72.727273	0.39194	29.34	61.2338623	2.8064	32.425	8.55188141
0.46725	31.63	51.364366	0.49	32.1	48.9795918	2.6344	37.38	9.1102338	0.41	35.33	58.536585	0.49451	30.51	48.5328911	3.0009	33.12	7.99760072
0.65494	32.6	36.6445781	0.67	33.02	35.8208955	2.88	37.74	8.3333333	0.52	36.37	46.153846	0.67093	31.7	35.7712429	3.8122	35.67	6.29557736
1.0717	34.01	22.3943268	1.09	34.19	22.0183486	4.2151	39.67	5.6938151	0.66	37.48	36.363636	1.1064	33.1	21.691974	4.6875	38.14	5.12
2.8346	36.61	8.46680308	2.95	37.53	8.13559322	7.6518	45.3	3.1365169	0.84	38.53	28.571429	2.938	35.6	8.16882233	6.1147	42.09	3.9249677
5.4084	39.21	4.4375416	5.52	40.55	4.34782609	10.754	50.47	2.2317277	1.08	39.73	22.222222	6.2215	37.38	3.85759061	6.2256	42.1	3.85505012
7.722	41.39	3.10800311	7.83	43.6	3.0651341	13.081	54.17	1.8347221	1.82	42.73	13.186813	7.0051	37.49	3.42607529	9.3947	49.93	2.55463187
11.654	44.11	2.05937875	11.72	50.72	2.04778157				2.83	46.81	8.4805654						
									3.88	50.81	6.185567						

Airplane（512 ×512 ×24）

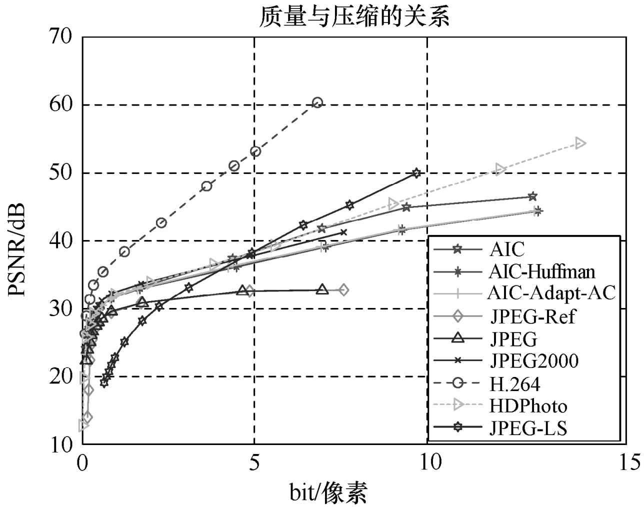




AIC、JPEG、JPEG 2000、HD Photo、H.264 和 JPEG LS 的仿真结果

M－AIC			JPEG2000			HD photo			H.264			JPEG-baseline			JPEG-LS		
比特率/(bit/像素)	PSNR/dB	压缩比（CR）	比特率/(bit/像素)	PSNR/dB	压缩比（CR）	比特率/(bit/像素)	PSNR/dB	压缩比（CR）	比特率/(bit/像素)	PSNR/dB	压缩比（CR）	比特率/(bit/像素)	PSNR/dB	压缩比（CR）	比特率/(bit/像素)	PSNR/dB	压缩比（CR）
0.1143	23.48	209.973753	0.16	26.79	150	0.023956	13.63	1001.8367	0.07	25.01	342.85714	0.15607	16.5	153.777151	0.79056	22.47	30.3582271
0.1517	25.09	158.206987	0.22	28.59	109.090909	0.050842	20.58	472.05067	0.12	27.98	200	0.17813	19.68	134.73306	1.007	24.76	23.8331678
0.1931	26.32	124.287934	0.26	28.98	92.3076923	0.12689	26.34	189.1402	0.2	30.9	120	0.21967	23.47	109.254791	1.3376	28.16	17.9425837
0.2228	27.96	107.719928	0.31	29.86	77.4193548	0.30249	30.28	79.341466	0.34	33.8	70.588235	0.25955	24.89	92.4677326	1.6153	30.55	14.8579211
0.2788	28.98	86.0832138	0.41	31.45	58.5365854	0.73041	34.57	32.858258	0.59	36.84	40.677966	0.31259	26.37	76.777888	2.1247	33.65	11.2957123
0.3704	30.38	64.7948164	0.46	31.91	52.173913	1.01	35.77	23.762376	1.01	40.09	23.762376	0.41	27.95	58.5365854	3.4128	38.825	7.0323488
0.44	31.32	54.5454545	0.58	33	41.3793103	1.9229	38.91	12.481148	1.68	43.46	14.285714	0.45526	28.5	52.7171287	4.6841	42.46	5.1237164
0.5542	32.29	43.3056658	0.76	34.18	31.5789474	2.9306	41.06	8.1894493	2.7	48.11	8.8888889	0.58221	29.73	41.2222394	5.7483	45.28	4.17514743
0.7413	33.61	32.3755565	1.13	36.1	21.2389381	5.7585	45.7	4.167752	4.02	52.88	5.9701493	0.76318	30.8	31.447365	7.429	49.915	3.23058285
1.1075	35.4	21.6704289	2.34	39.24	10.2564103	8.7902	50.54	2.7303133	5.78	59.88	4.1522491	1.1266	32.22	21.3030357			
2.2782	38.21	10.5346326	4.06	41.79	5.91133005	11.135	54.2	2.155366				2.3382	34.3	10.2643059			
3.9669	40.28	6.05006428	6.67	45.18	3.5982009							4.0624	35.32	5.90783773			
5.8615	41.96	4.09451506										6.671	35.82	3.59766152			
9.7702	44.25	2.4564492															

Peppers（512×512×24）

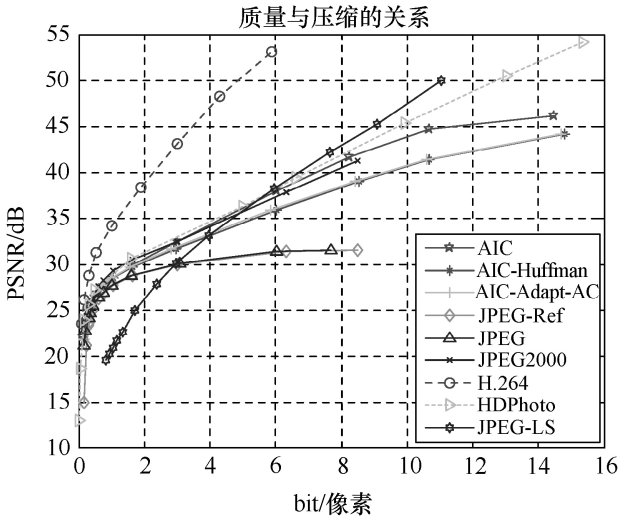




AIC、JPEG、JPEG 2000、HD Photo、H.264 和 JPEG LS 的仿真结果

M－AIC			JPEG2000			HD photo			H.264			JPEG-baseline			JPEG-LS		
比特率/(bit/像素)	PSNR/dB	压缩比(CR)	比特率/(bit/像素)	PSNR/dB	压缩比(CR)	比特率/(bit/像素)	PSNR/dB	压缩比(CR)	比特率/(bit/像素)	PSNR/dB	压缩比(CR)	比特率/(bit/像素)	PSNR/dB	压缩比(CR)	比特率/(bit/像素)	PSNR/dB	压缩比(CR)
0.12134	23.21	197.79133	0.17	26.87	141.176471	0.024567	12.66	976.92026	0.09	26.27	266.66667	0.15656	13.87	153.295861	0.63406	19.025	37.8513074
0.15509	24.88	154.748856	0.21	27.35	114.285714	0.039886	19.74	601.71489	0.13	28.91	184.61538	0.17413	17.97	137.828059	0.71481	19.9	33.5753557
0.19623	25.93	122.305458	0.26	28.24	92.3076923	0.13266	25.66	180.91361	0.21	31.3	114.28571	0.21042	22.31	114.057599	0.77441	20.668	30.9913353
0.224	27.3	107.142857	0.3	28.95	80	0.214	27.58	112.14953	0.34	33.44	70.588235	0.26248	24.61	91.4355379	0.85931	21.63	27.9293852
0.27603	28.18	86.947071	0.4	29.83	60	0.32788	29.22	73.197511	0.61	35.4	39.344262	0.29678	25.41	80.867983	0.95502	22.656	25.1303638
0.36841	29.14	65.1448115	0.46	30.45	52.173913	0.94431	32.09	25.415383	1.24	38.26	19.354839	0.40015	27.04	59.9775084	1.2293	25	19.5233059
0.4447	29.8	53.9689678	0.58	31.08	41.3793103	1.9774	33.79	12.13715	2.29	42.55	10.480349	0.46396	27.7	51.7285973	1.7383	28.165	13.8065926
0.56726	30.5	42.3086415	0.85	32.13	28.2352941	3.824	36.38	6.2761506	3.62	48.06	6.6298343	0.58496	28.45	41.0284464	2.2233	30.24	10.7947645
0.82291	31.42	29.1647932	1.72	33.65	13.9534884	5.5888	39.19	4.2943029	4.42	51.02	5.4298643	0.84818	29.4	28.2958806	3.1051	33.1	7.72921967
1.5903	32.83	15.0914922	4.85	37.59	4.94845361	8.9885	45.31	2.6700784	5.02	53.08	4.7808765	1.7238	30.8	13.9227289	4.9356	38.24	4.86263068
4.2245	36.09	5.6811457	7.59	41.22	3.16205534	12.115	50.51	1.9810153	6.81	60.33	3.5242291	4.8457	32.46	4.95284479	6.3958	42.127	3.75246255
6.9234	39.12	3.4665049				14.428	54.24	1.6634322				7.5877	32.8	3.16301383	7.7596	45.128	3.09294294
9.193	41.44	2.61068204													9.7025	49.896	2.47358928
13.128	44.25	1.82815356															

Saiboat on lake（512 × 512 × 24）

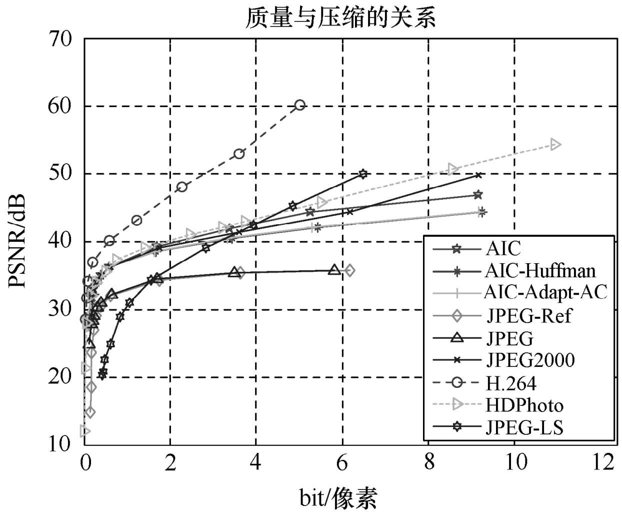


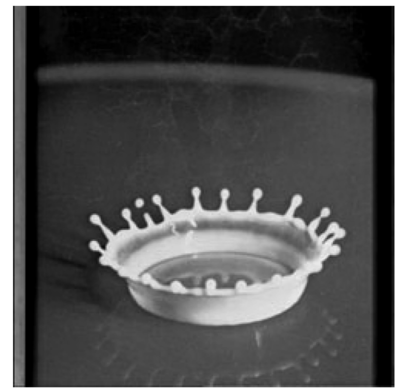


AIC、JPEG、JPEG 2000、HD Photo、H.264 和 JPEG LS 的仿真结果

M－AIC			JPEG2000			HD photo			H.264			JPEG-Baseline			JPEG-LS		
比特率/(bit/像素)	PSNR/dB	压缩比（CR）	比特率/(bit/像素)	PSNR/dB	压缩比（CR）	比特率/(bit/像素)	PSNR/dB	压缩比（CR）	比特率/(bit/像素)	PSNR/dB	压缩比（CR）	比特率/(bit/像素)	PSNR/dB	压缩比（CR）	比特率/(bit/像素)	PSNR/dB	压缩比（CR）
0.14911	21.83	160.955	0.16	23.34	150	0.02417	12.99	992.96649	0.09	23.52	266.66667	0.15973	14.96	150.253553	0.83322	19.526	28.8039173
0.20798	22.92	115.395711	0.24	24.83	100	0.039459	18.68	608.22626	0.17	26.16	141.17647	0.23822	21.15	100.747208	0.92972	20.164	25.8142236
0.27719	23.88	86.5832101	0.32	25.6	75	0.17169	23.9	139.78683	0.3	28.7	80	0.31549	23.3	76.0721417	1.0504	20.81	22.8484387
0.32007	25.11	74.9835973	0.34	25.76	70.5882353	0.3	25.64	80	0.54	31.26	44.444444	0.33929	23.75	70.7359486	1.177	21.705	20.3908241
0.41559	25.94	57.749224	0.43	26.35	55.8139535	0.48898	27.15	49.081762	1.01	34.08	23.762376	0.43073	24.9	55.7193602	1.3291	22.592	18.057332
0.58453	26.97	41.0586283	0.61	27.58	39.3442623	1.5885	30.53	15.108593	1.9	38.38	12.631579	0.60791	26.19	39.4795282	1.7169	24.873	13.9786825
0.73557	27.66	32.6277581	0.76	28.25	31.5789474	4.9777	36.31	4.8215039	3	43.15	8	0.75742	26.84	31.6865147	2.3803	27.81	10.0827627
1.0123	28.5	23.7083868	1.04	29.22	23.0769231	6.6033	39.35	3.6345464	4.29	48.2	5.5944056	1.0437	27.66	22.9951135	2.9586	30.09	8.11194484
1.5753	29.69	15.2351933	1.64	30.5	14.6341463	9.9368	45.34	2.4152645	5.89	53.08	4.0747029	1.6442	28.71	14.5967644	3.9549	33.07	6.06842145
2.8327	31.68	8.47248208	3.01	32.44	7.97342193	13.029	50.46	1.8420447				3.0138	29.96	7.9633685	5.9442	38.28	4.03754921
5.836	35.82	4.11240576	6.33	37.77	3.79146919	15.346	54.17	1.5639255				6.334	31.37	3.78907483	7.6394	42.15	3.14160798
8.4369	39.06	2.84464673	8.5	41.2	2.82352941							8.5009	31.49	2.82323048	9.0952	45.15	2.63875451
10.61	41.37	2.26201697													11.027	49.9	2.17647592
14.704	44.13	1.63220892															

Splash（512×512×24）

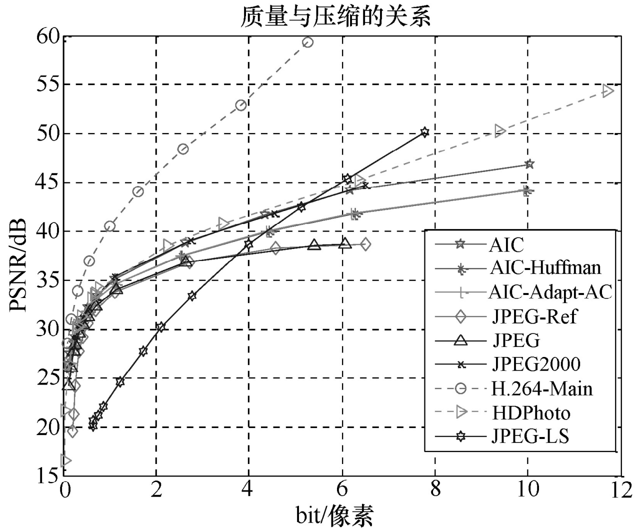




AIC、JPEG、JPEG 2000、HD Photo、H.264 和 JPEG LS 的仿真结果

M－AIC			JPEG2000			HD photo			H.264			JPEG-Baseline			JPEG-LS		
比特率/(bit/像素)	PSNR/dB	压缩比(CR)	比特率/(bit/像素)	PSNR/dB	压缩比(CR)	比特率/(bit/像素)	PSNR/dB	压缩比(CR)	比特率/(bit/像素)	PSNR/dB	压缩比(CR)	比特率/(bit/像素)	PSNR/dB	压缩比(CR)	比特率/(bit/像素)	PSNR/dB	压缩比(CR)
0.10254	27.48	234.055003	0.15	31.26	160	0.02359	12.08	339.13	0.05	28.5	480	0.1528	14.89	157.068063	0.44095	20.251	54.4279397
0.10785	29	222.531293	0.19	31.81	126.315789	0.030548	21.35	261.88	0.07	31.62	342.85714	0.16653	18.5	144.118177	0.46967	20.798	51.0997083
0.14493	31	165.597185	0.23	32.9	104.347826	0.079132	28.03	101.1	0.1	34.1	240	0.18524	23.67	129.56165	0.47461	22.537	50.5678346
0.18427	32.23	130.243664	0.3	34	80	0.17438	32.01	45.877	0.22	36.95	109.09091	0.23126	26.9	103.779296	0.61774	24.786	38.8512967
0.27219	33.85	88.1737022	0.39	34.98	61.5384615	0.27927	33.98	28.646	0.6	40.11	40	0.29617	29.05	81.034541	0.84161	28.86	28.5167714
0.37198	34.86	64.5195978	0.62	36.24	38.7096774	0.5	35.55	16	1.23	43.09	19.512195	0.39349	30.5	60.9926555	1.0721	31.039	22.3859715
0.5983	36.27	40.1136554	1.76	38.96	13.6363636	0.75558	37.24	10.588	2.28	48.03	10.526316	0.62057	31.97	38.6741222	1.5586	34.225	15.3984345
1.6328	38.48	14.6986771	3.62	41.3	6.62983425	1.4252	39.06	5.6134	3.62	52.96	6.6298343	1.7577	34.38	13.6542072	2.8232	39	8.50099178
3.3429	40.4	7.17939514	6.17	44.26	3.8897893	2.474	40.88	3.2337	5.02	60.06	4.7808765	3.6233	35.27	6.62379599	3.9319	42.397	6.10391922
5.3046	42.07	4.52437507	9.16	49.65	2.62008734	3.7391	42.83	2.1395				6.1726	35.76	3.88815086	4.8446	45.212	4.95396937
9.1562	44.37	2.62117472				5.5261	45.65	1.4477							6.4945	49.899	3.6954346
						8.5522	50.56	0.93543									
						10.932	54.23	0.73183									

Couple（256 × 256 × 24）

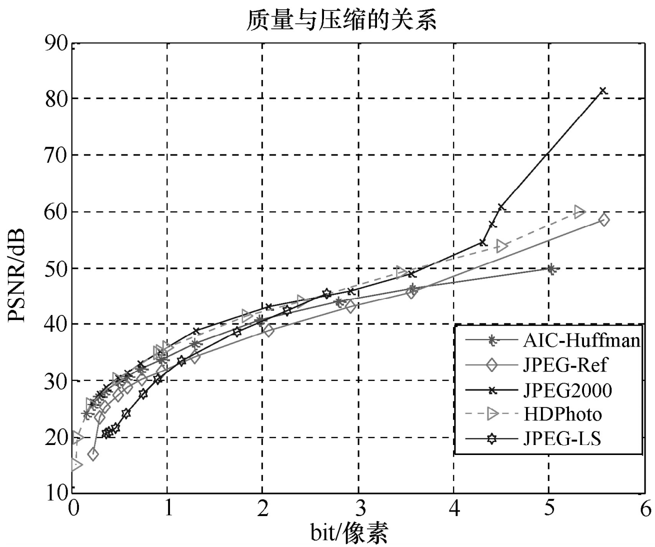




AIC、JPEG、JPEG 2000、HD Photo、H. 264 和 JPEG LS 的仿真结果

M - AIC			JPEG2000			HD photo			H. 264			JPEG-Baseline			JPEG-LS		
比特率/(bit/像素)	PSNR/dB	压缩比(CR)	比特率/(bit/像素)	PSNR/dB	压缩比(CR)	比特率/(bit/像素)	PSNR/dB	压缩比(CR)	比特率/(bit/像素)	PSNR/dB	压缩比(CR)	比特率/(bit/像素)	PSNR/dB	压缩比(CR)	比特率/(bit/像素)	PSNR/dB	压缩比(CR)
0. 12695	25. 82	189. 050807	0. 23	28. 24	104. 347826	0. 043945	16. 49	546. 137217	0. 1	28. 52	240	0. 22009	19. 49	109. 0463	0. 63611	20. 094	37. 7293235
0. 1582	26. 57	151. 7067	0. 27	29. 23	88. 8888889	0. 050049	21. 7	479. 530061	0. 18	31. 05	133. 333333	0. 23071	21. 22	104. 0267	0. 6593	20. 63	36. 4022448
0. 18762	28. 03	127. 918132	0. 35	30. 21	68. 5714286	0. 11401	26. 41	210. 50785	0. 33	33. 9	72. 7272727	0. 26501	24. 09	90. 56262	0. 76807	21. 068	31. 247152
0. 23975	28. 94	100. 104275	0. 43	31. 12	55. 8139535	0. 30103	30. 11	79. 7262731	0. 57	36. 95	42. 1052632	0. 349	27. 72	68. 767908	0. 8634	22. 089	27. 7970813
0. 32825	30. 09	73. 1150038	0. 54	32. 13	44. 4444444	0. 70605	33. 62	33. 9919269	1	40. 5	24	0. 42932	29. 14	55. 902357	1. 219	24. 566	19. 6882691
0. 40479	30. 84	59. 2900022	0. 71	33. 23	33. 8028169	0. 79602	34. 09	30. 1499962	1. 63	44. 07	14. 7239264	0. 54102	30. 63	44. 360652	1. 7167	27. 642	13. 9803111
0. 51587	31. 76	46. 5233489	1. 13	35. 39	21. 2389381	2. 2512	38. 46	10. 6609808	2. 59	48. 35	9. 26640927	0. 70703	31. 97	33. 944811	2. 106	30. 248	11. 3960114
0. 70776	32. 9	33. 9097999	2. 74	39. 07	8. 75912409	3. 4338	40. 72	6. 98934125	3. 84	52. 78	6. 25	1. 1259	33. 81	21. 31628	2. 7726	33. 398	8. 65613504
1. 1217	34. 58	21. 3960952	4. 56	41. 78	5. 26315789	6. 3629	45. 18	3. 77186503	5. 28	59. 23	4. 54545455	2. 7365	36. 76	8. 7703271	4. 002	38. 621	5. 9970015
2. 5236	37. 5	9. 51022349	6. 5	44. 73	3. 69230769	9. 3762	50. 16	2. 55967236				4. 5638	38. 21	5. 2587756	5. 1161	42. 392	4. 69107328
4. 3724	39. 9	5. 48897631				11. 708	54. 24	2. 04988042				6. 5031	38. 68	3. 6905476	6. 1143	45. 309	3. 92522447
6. 2196	41. 75	3. 85876905													7. 7755	50. 021	3. 08661822
9. 9216	44. 11	2. 41896468															

Cameraman（256 ×256 ×8）

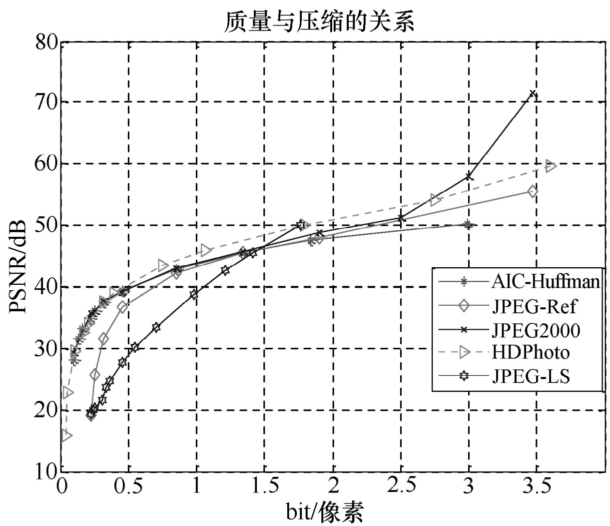




AIC、JPEG、JPEG 2000、HD Photo 和 JPEG LS 的仿真结果

M－AIC			JPEG2000			HD photo			JPEG-Baseline			JPEG-LS		
比特率/(bit/像素)	PSNR/dB	压缩比(CR)	比特率/(bit/像素)	PSNR/dB	压缩比(CR)	比特率/(bit/像素)	PSNR/dB	压缩比(CR)	比特率/(bit/像素)	PSNR/dB	压缩比(CR)	比特率/(bit/像素)	PSNR/dB	压缩比(CR)
0.16101	24.19	49.686	0.23	26.15	34.783	0.044067	15.08	181.54	0.22766	16.91	35.14	2.6743	45.5	2.9914
0.23462	25.69	34.098	0.3	27.73	26.667	0.062256	19.97	128.5	0.29504	23.39	27.115	2.2651	42.46	3.5318
0.27844	27.09	28.731	0.36	28.73	22.222	0.19922	25.73	40.157	0.35669	25.4	22.428	1.7388	38.73	4.601
0.35815	28.19	22.337	0.49	30.4	16.327	0.47461	30.18	16.856	0.48901	27.49	16.359	1.1558	33.46	6.9218
0.48804	29.71	16.392	0.59	31.33	13.559	0.939	35.14	8.5156	0.59253	28.76	13.501	0.91016	30.334	8.7897
0.59216	30.74	13.51	0.73	32.97	10.959	1.0066	35.82	7.9476	0.73438	30.15	10.894	0.75732	27.68	10.564
0.73218	32.05	10.926	0.94	35.16	8.5106	1.832	41.45	4.3667	0.94214	31.74	8.4913	0.56799	24.07	14.085
0.94409	33.84	8.4738	1.3	38.81	6.1538	2.4185	44.15	3.3079	1.2992	34.15	6.1577	0.45667	21.68	17.518
1.2936	36.44	6.1844	2.06	43.15	3.8835	3.4471	49.24	2.3208	2.0635	38.8	3.877	0.39392	20.92	20.309
1.9724	40.75	4.0559	2.92	45.97	2.7397	4.5032	53.83	1.7765	2.9231	43.06	2.7368	0.36206	20.66	22.096
2.8062	43.93	2.8509	3.56	48.87	2.2472	5.3182	59.83	1.5043	3.5594	45.6	2.2475			
3.5699	46.48	2.2409	4.3	54.66	1.8605				5.5785	58.53	1.4341			
5.0234	49.98	1.5925	4.4	57.86	1.8182									
			4.5	60.93	1.7778									
			5.57	81.52	1.4363									

Man（256 ×256 ×8）

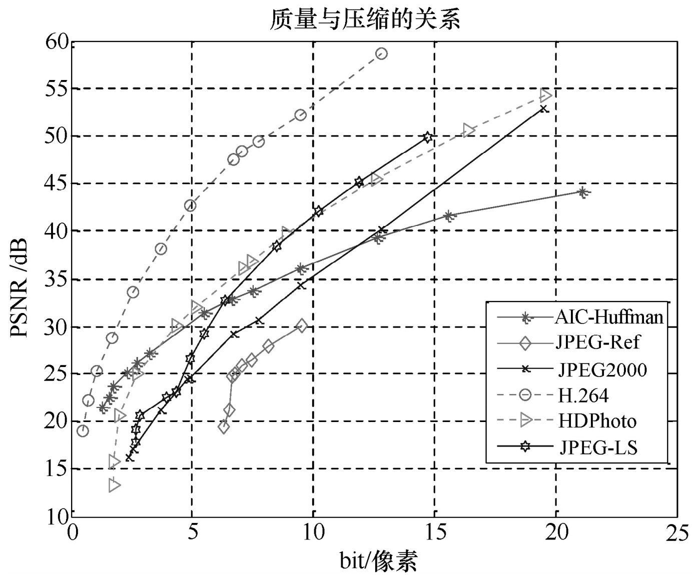

质量与压缩的关系
PSNR/dB
bit/像素
80
70
60
50
40
30
20
10
0
0.5
1
1.5
2
2.5
3
3.5
AIC-Huffman
JPEG-Ref
JPEG2000
HDPhoto
JPEG-LS



AIC、JPEG、JPEG 2000、HD Photo 和 JPEG LS 的仿真结果

M－AIC			JPEG2000			HD photo			JPEG-Baseline			JPEG-LS		
比特率 /(bit/像素)	PSNR /dB	压缩比 (CR)	比特率 /(bit/像素)	PSNR /dB	压缩比 (CR)	比特率 /(bit/像素)	PSNR /dB	压缩比 (CR)	比特率 /(bit/像素)	PSNR /dB	压缩比 (CR)	比特率 /(bit/像素)	PSNR /dB	压缩比 (CR)
0. 097046	28. 03	82. 435	0. 1	29. 26	80	0. 044678	15. 76	179. 06	0. 22656	19. 2	35. 31	0. 22131	19. 373	36. 148
0. 13342	31. 59	59. 96	0. 23	35. 71	34. 783	0. 055298	22. 77	144. 67	0. 25427	25. 63	31. 462	0. 24207	19. 88	33. 049
0. 16101	33. 25	49. 686	0. 25	36. 05	32	0. 10913	29. 53	73. 306	0. 31604	31. 55	25. 313	0. 25549	20. 275	31. 312
0. 21375	34. 9	37. 428	0. 32	37. 66	25	0. 17114	32. 66	46. 745	0. 45386	36. 67	17. 627	0. 31226	21. 51	25. 62
0. 25049	36. 01	31. 938	0. 45	39. 21	17. 778	0. 20679	34. 04	38. 687	0. 85303	42. 33	9. 3784	0. 33435	23. 51	23. 927
0. 31982	37. 47	25. 014	0. 85	43. 15	9. 4118	0. 32117	37. 6	24. 909	1. 3411	45. 52	5. 9654	0. 3623	24. 64	22. 081
0. 45251	39. 42	17. 679	1. 34	45. 66	5. 9701	0. 38928	38. 87	20. 551	1. 9014	47. 95	4. 2075	0. 4585	27. 79	17. 448
0. 85022	42. 89	9. 4093	1. 9	48. 76	4. 2105	0. 75366	43. 48	10. 615	3. 4674	55. 47	2. 3072	0. 54395	30. 225	14. 707
1. 332	45. 5	6. 0059	2. 5	51. 29	3. 2	1. 0673	45. 86	7. 4958				0. 70203	33. 43	11. 396
1. 8369	47. 49	4. 3551	3	58. 02	2. 6667	1. 7921	49. 99	4. 464				0. 9845	38. 77	8. 126
2. 9894	50. 28	2. 6761	3. 47	71. 41	2. 3055	2. 7471	54. 05	2. 9122				1. 2136	42. 598	6. 5918
						3. 6007	59. 7	2. 2218				1. 4169	45. 45	5. 6462
												1. 7649	50	4. 5329

Lena（32 ×32 ×24）

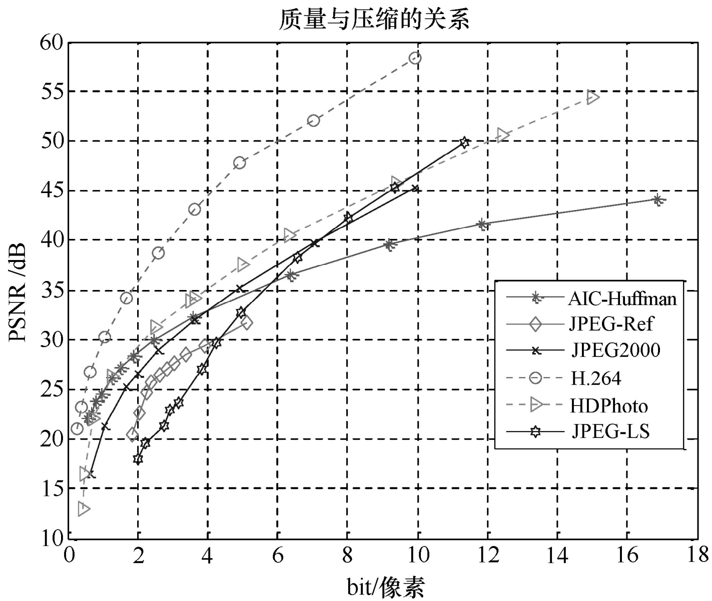

质量与压缩的关系
PSNR /dB
bit/像素
60
55
50
45
40
35
30
25
20
15
10
0
5
10
15
20
25
AIC-Huffman
JPEG-Ref
JPEG2000
H.264
HDPhoto
JPEG-LS



AIC、JPEG、JPEG 2000、HD Photo、H.264 和 JPEG LS 的仿真结果

M－AIC			JPEG2000			HD photo			H.264			JPEG-Baseline			JPEG-LS		
比特率/(bit/像素)	PSNR/dB	压缩比(CR)	比特率/(bit/像素)	PSNR/dB	压缩比(CR)	比特率/(bit/像素)	PSNR/dB	压缩比(CR)	比特率/(bit/像素)	PSNR/dB	压缩比(CR)	比特率/(bit/像素)	PSNR/dB	压缩比(CR)	比特率/(bit/像素)	PSNR/dB	压缩比(CR)
1.3	21.44	18.4615385	2.4	16.25	10	1.7422	13.36	13.7756859	0.47	18.94	51.0638298	6.29	19.47	3.8155803	2.6875	17.788	8.93023256
1.58	22.48	15.1898734	2.59	17.09	9.26640927	1.75	15.82	13.7142857	0.72	22.2	33.3333333	6.54	21.22	3.6697248	2.6953	19.164	8.90438912
1.78	23.62	13.4831461	3.69	21.23	6.50406504	2.0078	20.65	11.9533818	1.09	25.35	22.0183486	6.66	24.76	3.6036036	2.8359	20.548	8.46292182
2.32	25.18	10.3448276	4.8	24.37	5	2.7188	24.99	8.82742386	1.72	28.8	13.9534884	6.79	24.94	3.5346097	3.8516	20.838	6.23117665
2.75	26.09	8.72727273	4.91	24.56	4.88798371	4.3516	30.08	5.5152128	2.59	33.56	9.26640927	7.08	25.83	3.3898305	3.9453	22.566	6.08318759
3.24	27.15	7.40740741	6.69	29.21	3.58744395	5.1406	32.05	4.66871571	3.69	38.14	6.50406504	7.45	26.44	3.2214765	4.3203	23.15	5.55516978
5.52	31.43	4.34782609	7.72	30.71	3.10880829	7.1	36.06	3.38028169	4.91	42.66	4.88798371	8.16	27.9	2.9411765	4.9531	26.653	4.84545032
6.68	32.85	3.59281437	9.5	34.4	2.52631579	7.4844	36.76	3.20666987	6.69	47.45	3.58744395	9.54	30.03	2.5157233	5.5	29.283	4.36363636
7.52	33.81	3.19148936	12.84	40.2	1.86915888	8.8984	39.82	2.69711409	7.06	48.38	3.39943343				6.3438	32.735	3.78322141
9.48	36.12	2.53164557	19.5	52.85	1.23076923	12.602	45.47	1.90445961	7.72	49.4	3.10880829				8.5234	38.47	2.81577774
12.66	39.35	1.8957346				16.391	50.6	1.46421817	9.5	52.17	2.52631579				10.234	42.13	2.3451241
15.59	41.59	1.53944836				19.539	54.29	1.22831261	12.84	58.55	1.86915888				11.883	45.1	2.019692
21.11	44.16	1.13690194													14.719	49.89	1.63054555

Lena（64 ×64 ×24）

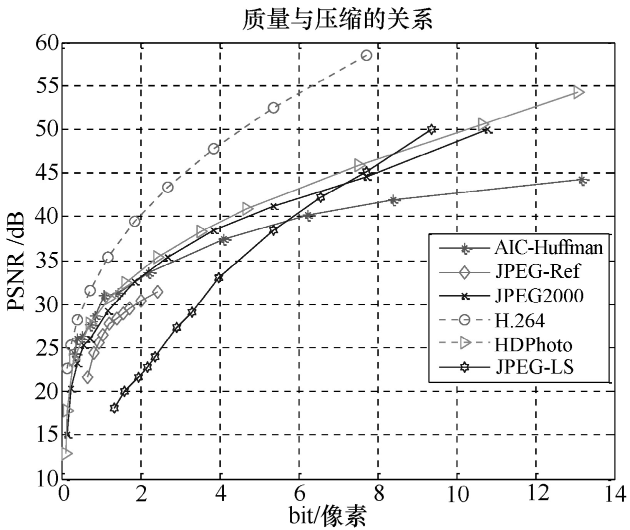

质量与压缩的关系
PSNR /dB
bit/像素
60
55
50
45
40
35
30
25
20
15
10
0
2
4
6
8
10
12
14
16
18
AIC-Huffman
JPEG-Ref
JPEG2000
H.264
HDPhoto
JPEG-LS



AIC、JPEG、JPEG 2000、HD Photo、H. 264 和 JPEG LS 的仿真结果

M－AIC			JPEG2000			HD photo			H. 264			JPEG-Baseline			JPEG-LS		
比特率/(bit/像素)	PSNR/dB	压缩比(CR)	比特率/(bit/像素)	PSNR/dB	压缩比(CR)	比特率/(bit/像素)	PSNR/dB	压缩比(CR)	比特率/(bit/像素)	PSNR/dB	压缩比(CR)	比特率/(bit/像素)	PSNR/dB	压缩比(CR)	比特率/(bit/像素)	PSNR/dB	压缩比(CR)
0. 55	22. 1	43. 6363636	0. 66	16. 51	36. 3636364	0. 45508	13. 06	52. 7379801	0. 25	21. 07	96	1. 83	20. 45	13. 114754	1. 998	17. 96	12. 012012
0. 67	22. 65	35. 8208955	1. 05	21. 31	22. 8571429	0. 48047	16. 53	49. 9510896	0. 41	23. 21	58. 5365854	2. 05	22. 69	11. 707317	2. 2109	19. 545	10. 8553078
0. 8	23. 77	30	1. 7	25. 28	14. 1176471	0. 74023	22. 11	32. 4223552	0. 66	26. 8	36. 3636364	2. 24	24. 69	10. 714286	2. 752	21. 298	8. 72093023
0. 96	24. 57	25	2	26. 63	12	1. 2852	26. 27	18. 6741363	1. 05	30. 25	22. 8571429	2. 4	25. 73	10	2. 9414	22. 936	8. 15937989
1. 26	26. 09	19. 047619	2. 61	28. 97	9. 1954023	2. 502	31. 26	9. 59232614	1. 7	34. 14	14. 1176471	2. 62	26. 41	9. 1603053	3. 1797	23. 686	7. 54788188
1. 53	27. 14	15. 6862745	3. 64	31. 99	6. 59340659	3. 6328	34. 24	6. 60647434	2. 61	38. 73	9. 1954023	2. 84	27. 03	8. 4507042	3. 8184	27. 08	6. 28535512
1. 88	28. 29	12. 7659574	4. 92	35. 26	4. 87804878	5. 0078	37. 57	4. 79252366	3. 64	43. 11	6. 59340659	3. 06	27. 61	7. 8431373	4. 2676	29. 61	5. 6237698
2. 47	29. 92	9. 71659919	7. 03	39. 81	3. 41394026	6. 3262	40. 42	3. 79374664	4. 92	47. 74	4. 87804878	3. 4	28. 49	7. 0588235	4. 9766	32. 756	4. 82256963
3. 59	32. 31	6. 68523677	9. 92	45. 37	2. 41935484	9. 3984	45. 7	2. 55362615	7. 03	52. 02	3. 41394026	3. 92	29. 43	6. 122449	6. 5918	38. 26	3. 64088716
6. 38	36. 56	3. 76175549				12. 43	50. 56	1. 93081255	9. 92	58. 28	2. 41935484	5. 12	31. 71	4. 6875	8. 0254	42. 22	2. 99050515
9. 17	39. 62	2. 6172301				14. 994	54. 32	1. 60064026							9. 3516	45. 24	2. 56640575
11. 85	41. 68	2. 02531646													11. 346	49. 863	2. 11528292
16. 85	44. 18	1. 42433234															

Lena（128 ×128 ×24）

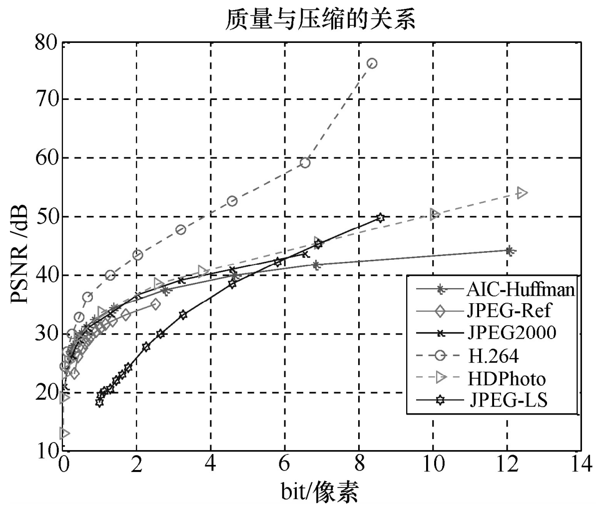




AIC、JPEG、JPEG 2000、HD Photo、H.264 和 JPEG LS 的仿真结果

M－AIC			JPEG2000			HD photo			H.264			JPEG-Baseline			JPEG-LS		
比特率/(bit/像素)	PSNR/dB	压缩比(CR)	比特率/(bit/像素)	PSNR/dB	压缩比(CR)	比特率/(bit/像素)	PSNR/dB	压缩比(CR)	比特率/(bit/像素)	PSNR/dB	压缩比(CR)	比特率/(bit/像素)	PSNR/dB	压缩比(CR)	比特率/(bit/像素)	PSNR/dB	压缩比(CR)
0.35	24.27	68.5714286	0.16	15.09	150	0.12646	12.92	189.783331	0.16	22.59	150	0.66	21.69	36.363636	1.3438	18.105	17.8598006
0.41	26.08	58.5365854	0.25	20.24	96	0.15381	17.78	156.036669	0.25	25.3	96	0.82	24.47	29.268293	1.5869	20.04	15.1238263
0.52	26.27	46.1538462	0.41	23.29	58.5365854	0.32178	23.6	74.5851203	0.41	28.26	58.5365854	0.95	25.64	25.263158	1.9668	21.63	12.2025625
0.72	27.65	33.3333333	0.58	25.21	41.3793103	0.73828	27.8	32.5079915	0.71	31.51	33.8028169	1.05	26.46	22.857143	2.1777	22.753	11.0208018
0.86	28.59	27.9069767	0.71	25.99	33.8028169	1.2	30.65	20	1.18	35.32	20.3389831	1.22	27.7	19.672131	2.3687	23.99	10.13214
1.08	30.99	22.2222222	1.18	29.27	20.3389831	1.6196	32.49	14.8184737	1.85	39.53	12.972973	1.39	28.41	17.266187	2.9106	27.35	8.24572253
1.44	31.3	16.6666667	1.85	32.52	12.972973	2.4238	35.39	9.90180708	2.7	43.46	8.88888889	1.57	28.96	15.286624	3.3203	29.07	7.22826251
2.21	33.65	10.8597285	2.7	35.38	8.88888889	3.5337	38.32	6.79174803	3.85	47.76	6.23376623	1.74	29.58	13.793103	3.9902	33.057	6.0147361
4.11	37.42	5.83941606	3.85	38.4	6.23376623	4.6812	40.94	5.12689054	5.38	52.4	4.46096654	2.01	30.33	11.940299	5.3867	38.414	4.45541797
6.23	40.13	3.85232745	5.38	41.18	4.46096654	7.5483	45.88	3.17952387	7.74	58.52	3.10077519	2.44	31.42	9.8360656	6.5757	42.195	3.64980154
8.41	41.97	2.85374554	7.74	44.54	3.10077519	10.65	50.51	2.25352113							7.7295	45.2	3.10498739
13.18	44.26	1.82094082	10.8	49.95	2.22222222	13.076	54.18	1.83542368							9.3696	49.926	2.56147541

Lena（256 ×256 ×24）





AIC、JPEG、JPEG 2000、HD Photo、H.264 和 JPEG LS 的仿真结果

M－AIC			JPEG2000			HD photo			H.264			JPEG-Baseline			JPEG-LS		
比特率/(bit/像素)	PSNR/dB	压缩比(CR)	比特率/(bit/像素)	PSNR/dB	压缩比(CR)	比特率/(bit/像素)	PSNR/dB	压缩比(CR)	比特率/(bit/像素)	PSNR/dB	压缩比(CR)	比特率/(bit/像素)	PSNR/dB	压缩比(CR)	比特率/(bit/像素)	PSNR/dB	压缩比(CR)
0.22	26.11	109.090909	0.09	21	266.666667	0.0448	12.96	535.714286	0.09	24.45	266.666667	0.33	23.15	72.727273	1.0103	18.206	23.7553202
0.25	26.92	96	0.16	23.83	150	0.062378	19.04	384.751034	0.16	26.92	150	0.45	26	53.333333	1.0494	19.455	22.8702115
0.32	27.85	75	0.27	26.37	88.8888889	0.11548	23.2	207.828195	0.27	29.87	88.8888889	0.55	27.46	43.636364	1.1318	20.041	21.2051599
0.44	29.2	54.5454545	0.46	28.98	52.173913	0.17786	25.22	134.937591	0.46	32.81	52.173913	0.63	28.33	38.095238	1.2928	20.583	18.5643564
0.52	30.09	46.1538462	0.7	31.08	34.2857143	0.44263	29.3	54.2213587	0.7	36.2	34.2857143	0.71	29	33.802817	1.4529	21.906	16.5186868
0.65	31.12	36.9230769	1.32	33.44	18.1818182	1.0736	33.69	22.3546945	1.32	39.94	18.1818182	0.78	29.52	30.769231	1.6145	22.988	14.8652834
0.9	32.45	26.6666667	2.06	36.77	11.6504854	2.6025	38.49	9.22190202	2.06	43.4	11.6504854	0.92	30.3	26.086957	1.7871	24.314	13.4295786
1.39	34.45	17.2661871	3.2	39.18	7.5	3.7887	40.68	6.33462665	3.2	47.76	7.5	1.04	30.94	23.076923	2.2773	27.645	10.5387959
2.79	37.6	8.60215054	4.6	41.02	5.2173913	6.9186	45.54	3.4689099	4.6	52.63	5.2173913	1.18	31.5	20.338983	2.6547	29.924	9.04056956
4.66	39.89	5.15021459	6.56	43.59	3.65853659	10.043	50.48	2.38972419	6.56	59.32	3.65853659	1.38	32.21	17.391304	3.2648	33.276	7.35113943
6.87	41.8	3.49344978				12.416	54.19	1.93298969	8.37	76.17	2.86738351	1.72	33.25	13.953488	4.6106	38.568	5.20539626
12.07	44.25	1.98840099										2.52	35.1	9.5238095	5.828	42.324	4.11805079
															6.9159	45.233	3.47026417
															8.5831	49.895	2.79619252

每幅图像的典型比较结果

Test Images	M – AIC			JPEG2000			HD photo			H. 264			JPEG- Baseline			JPEG- LS		
	比特率/(bit/像素)	PSNR/dB	压缩比(CR)	比特率/(bit/像素)	PSNR/dB	压缩比(CR)	比特率/(bit/像素)	PSNR/dB	压缩比(CR)	比特率/(bit/像素)	PSNR/dB	压缩比(CR)	比特率/(bit/像素)	PSNR/dB	压缩比(CR)	比特率/(bit/像素)	PSNR/dB	压缩比(CR)
Airplane (512×512×24)	1. 1075	35. 4	21. 670429	1. 13	36. 1	21. 238938	1. 01	35. 77	23. 762376	1. 01	40. 09	23. 762376	1. 1266	32. 22	21. 303036	1. 007	24. 76	23. 833168
Lena (512×512×24)	2. 8346	36. 61	8. 4668031	2. 95	37. 53	8. 1355932	2. 88	37. 74	8. 3333333	2. 83	46. 81	8. 4805654	2. 938	35. 6	8. 1688223	2. 8064	32. 425	8. 5518814
Peppers (512×512×24)	1. 5903	32. 83	15. 091492	1. 72	33. 65	13. 953488	1. 9774	33. 79	12. 13715	1. 24	38. 26	19. 354839	1. 7238	30. 8	13. 922729	1. 7383	28. 165	13. 806593
Sailboat on Lake (512×512×24)	0. 32007	25. 11	74. 983597	0. 32	25. 6	75	0. 3	25. 64	80	0. 3	28. 7	80	0. 31549	23. 3	76. 072142	0. 83322	19. 526	28. 803917
Splash (512×512×24)	3. 3429	40. 4	7. 1793951	3. 62	41. 3	6. 6298343	3. 7391	42. 83	2. 1395	3. 62	52. 96	6. 6298343	3. 6233	35. 27	6. 623796	3. 9319	42. 397	6. 1039192
Couple (256×256×24)	0. 70776	32. 9	33. 9098	0. 71	33. 23	33. 802817	0. 70605	33. 62	33. 991927	0. 57	36. 95	42. 105263	0. 70703	31. 97	33. 944811	0. 76807	21. 068	31. 247152
Lena (32×32×24)	6. 68	32. 85	3. 5928144	6. 69	29. 21	3. 5874439	7. 1	36. 06	3. 3802817	6. 69	47. 45	3. 5874439	6. 66	24. 76	3. 6036036	6. 3438	32. 735	3. 7832214
Lena (64×64×24)	3. 59	32. 31	6. 6852368	3. 64	31. 99	6. 5934066	3. 6328	34. 24	6. 6064743	3. 64	43. 11	6. 5934066	3. 92	29. 43	6. 122449	3. 8184	27. 08	6. 2853551
Lena (128×128×24)	1. 08	30. 99	22. 222222	1. 18	29. 27	20. 338983	1. 2	30. 65	20	1. 18	35. 32	20. 338983	1. 22	27. 7	19. 672131	1. 3438	18. 105	17. 859801
Lena (256×256×24)	0. 44	29. 2	54. 545455	0. 46	28. 98	52. 173913	0. 44263	29. 3	54. 221359	0. 46	32. 81	52. 173913	0. 45	26	53. 333333	1. 0103	18. 206	23. 75532
Cameraman (256×256×8)	0. 94409	33. 84	8. 4738	0. 94	35. 16	8. 5106	0. 939	35. 14	8. 5156				0. 94214	31. 74	8. 4913	0. 91016	30. 334	8. 7897
Man (256×256×8)	0. 31982	37. 47	25. 014	0. 32	37. 66	25	0. 32117	37. 6	24. 909				0. 31604	31. 55	25. 313	0. 31226	21. 51	25. 62

附录 G　面向高清视频编码的更高阶二维整数余弦变换

本附录基于 Madhu Peringassery Krishnan 的学位论文。

欲知详情，请参见 UTA 的 EE5359 课程网址 http：//www-ee. uta. edu/Dip/Courses/EE5359/index. html. 上的 Thesis/Project Title。

G.1　离散余弦变换与视频压缩

H. 264/AVC 和 AVS-video 是两个应用范围宽广的视频编码标准，涵盖范围从高端的专业摄像机和编辑系统到低端的移动应用。它们力求最大的压缩效率，而不牺牲视频质量。为此，它们定义了许多编码工具。变换编码是其中之一。

变换编码在另一个域（变换域）中表示信号/图像（当前是在时域/空域），变换域中信号/图像的大多数能量集中在少数的系数上。这样，在变换编码之后不重要的系数就会被丢弃以实现压缩。在图像/视频中 DCT-II（它把信号/图像表示为不同频率余弦函数的加权和）主要用于变换编码。

H. 264/AVC 和 AVS-video 利用 DCT-II（叫作整数余弦变换）的整数近似，通过仅仅执行定点算术运算并消除正向和逆向变换之间的失配来降低计算复杂度。所用的整数余弦变换的阶数（大小）不大（4×4 和 8×8）。对于标清和低分辨率视频，它们实现了最好的编码效率。但是通过使用更高阶的整数余弦变换（16×16 和 32×32）对高清视频可以实现更好的编码效率。因为高清视频正在变得越来越流行，不可避免地它们迟早会被融入到标准之中。为此目的，人们提出了许多高阶（16×16 和 32×32）整数余弦变换。但是针对这些 H. 264/AVC 和 AVS-video 的高阶整数余弦变换性能的比较研究仍尚未进行。

本附录的目的在于分析一些高阶 16×16 的整数余弦变换，在 H. 264/AVC 和 AVS-video 中实现他们，并进行它们性能的比较研究。

离散余弦变换 II 型（DCT-II）［I1］由 Ahmed，Natarajan，和 Rao 提出，是那么多图像和视频压缩标准的基本构件。由于它的能量会聚特性，DCT-II 被认为是接近最优 Karhunen-Loeve transform（KLT）变换的最好的次优正交变换。KLT 变换通过空间去相关将能量集中于少量的系数上。DCT-II 也是一个由正交基矢量组成的酉变换。

大小为 8 的 DCT-II 用于许多视频标准中，例如 H. 261，JPEG，MPEG-1、MPEG-2、H. 263、和 MPEG-4［B9，I15，S11］。典型的大小为 8 的二维 DCT-II 可

以分离为两个一维变换，首先沿着图像或视频帧的一条轴执行，然后再沿着前一过程结果的另一条轴执行，如图 G. 1。

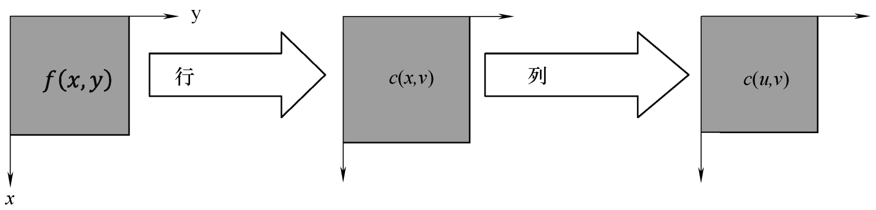


图 G. 1　DCT-Ⅱ的可分离特性

产生的余弦值难以近似成定点精度的整数，因而在实际应用中产生舍入误差。舍入误差会在计算中引入足够的误差，从而改变了变换的正交特性。

定点实现问题可以通过以下两种途径解决：（1）将余弦值近似为整数值或（2）通过保持余弦矩阵的相对幅度、关系和余弦值的对称性来创建一个可逆的整数余弦矩阵［I4、I6、I7、H4］。第一种方法导致编码器和译码器中变换的失配实现从而引入不当的结果。第二种方法是规定一个基于整数的变换，该变换可被精确地规定而且在编码器和译码器中易于实现。而且，对它们的设计要保证正变换和逆变换是完全可逆的，因而也是完全正交的。

G. 2　整数余弦变换

通过用整数代替 DCT-Ⅱ矩阵中的实值元素，同时保持矩阵元素间的相对幅值和正交关系，可以从 DCT-Ⅱ生成整数余弦变换（ICT）［I6］。整数变换系数使得计算过程烈度较轻，该过程实现类似于 DCT-Ⅱ的能量会聚作用。它可以用整数计算来实现而没有编码器和译码器之间的失配问题。而且，人们已经制定出实现它们的快速算法。编码器和译码器的典型框图见本书 15 页图 1. 4。

对于大于四的阶数，ICT 的正交性取决于变换矩阵的元素。由于这种限制，元素的幅值对于大的 ICT（16 阶 ICT）趋于相当大［I7］。通过利用二值对称原理，这促成了相互正交的 ICT 的发展［I4］。这样，变换矩阵的元素可以不受正交性约束而选出来。通常，元素的幅值被设计得很小（由 4 ~ 5bit 表示）以便于简单的实现。

最近几年中，ICT 已被广泛用于视频编码中，如 H. 264［H11］中的 4 阶和 8 阶 ICT，中国音视频编码标准（AVS）［AVS］的 8 阶 ICT，和 WM9-9［C2］。

视频图像/帧间的空间相关是彼此相邻的样值/像素间相关度的统计测度［B5］。与较低分辨率视频相比，高清视频具有更高的空间相关度［H26］。式（G. 1）定义的相关系数度量了邻近样值/像素的相关度，且可被认为是一个随机过

程［B6］。

$$r(n_1,n_2)=\frac{E[(x(n_1)-\mu_1)(x(n_2)-\mu_2)]}{\sigma_1\sigma_2} \tag{G.1}$$

式中，符号 n_1 和 n_2 表示附近的样值；$x(n_1)$ 和 $x(n_2)$ 表示样值 n_1 和 n_2 的强度值（其中 n_1 和 n_2 沿着行选取以便于 $r(n_1, n_2)$）给出水平相关系数）；μ_1、μ_2 和 σ_1、σ_2 分别表示均值和标准差；E 是期望运算符。表 G.1 给出针对不同分辨率的视频序列的预测残差计算出的相关系数。这里 r（1）是相邻像素的相关度，r（2）是间隔像素的相关度，μ_{r1} 和 μ_{r2} 是它们各自的均值，σ_{r1} 和 σ_{r2} 是它们各自的标准差。从表 G.1 可以看出，高清视频序列的 μ_{r1} 比较低分辨率视频相对较高，如宽视频图形阵列（WVGA）和宽 1/4 视频图形阵列（WQVGA）。而且，对于较低分辨率的视频序列 μ_{r2} 几乎为零，而对于高清序列它仍然很高。因而，高清视频一般而言会给出较高的空间相关性。通过使用较高阶的 ICT，高清视频的这种特性可被利用来实现更好的编码效率［H28］。

表 G.1　不同视频分辨率下相邻像素的相关度［I21］

测试序列	分辨率	r（1）		r（2）	
		μ_{r1}	σ_{r1}	μ_{r2}	σ_{r2}
Kimono	1920×1080（高清）	0.8673	0.1284	0.7311	0.1434
Parkscene		0.7431	0.1820	0.6695	0.1967
Cactus		0.8542	0.1692	0.7483	0.1245
Vidyo1	1280×720（高清）	0.7539	0.2401	0.4073	0.1842
Vidyo2		0.6643	0.1982	0.3060	0.1569
Vidyo3		0.5474	0.1125	0.3221	0.2923
Partyscene	832×480（WVGA）	0.4953	0.1598	0.2019	0.1757
BQmall		0.4517	0.2145	0.1966	0.2450
Basketballdrill		0.5594	0.1183	0.2301	0.1032
BQsquare	416×240（WQVGA）	0.3543	0.2935	0.0964	0.1722
Blowingbubbles		0.2879	0.1515	0.0473	0.1906
Basketballpass		0.2177	0.1784	0.0355	0.2098

对于 r（1）和 r（2），左列描述了预测残差相关系数的均值，右列描述了预测残差相关系数的标准差。［I16］的表 1 描述了像素的相关系数。

在随后的部分中会讨论从 DCT-Ⅱ到 ICT 的一般导出过程和将在 H.264/AVC［H29］和 AVS-vidwo［A59］中专门面向高清视频编码实现的三个特定的 2 维 16 阶 ICT。

DCT-Ⅱ［B2］把 N 长的矢量 $\boldsymbol{X}$ 映射到另一个同样长度的变换系数矢量 $\boldsymbol{X}$ 中（G.2）。

$$\boldsymbol{X}=[H]x \tag{G.2}$$

这里的矩阵［$\boldsymbol{H}$］叫作变换矩阵，定义为

$$H(k,n)=c_k\sqrt{\frac{2}{N}}\cos\left[\left(n+\frac{1}{2}\right)\frac{k\pi}{N}\right] \tag{G.3}$$

其中 k，$n=0$，1，…，$N-1$

$$c_k=\begin{cases}\frac{1}{\sqrt{2}}, k=0\\ 1, k\neq 0\end{cases}$$

矩阵［H］和［H］$^{\mathrm{T}}$的元素是无理数。这样，如果正变换和逆变换级联在一起实现的话，计算机将不能用有限比特的精度重建得到同样的数据。而且，如果正变换和反变换在不同浮点表示的不同机器上执行的话，误差将会很大。先缩放［H］和［H］$^{\mathrm{T}}$然后取整到最近的整数就能避免以上误差。但如果缩放因数很大，那么行（基矢量）的范数就会很高，计算复杂度也会提高。因而，具有小整数元素的正交矩阵［H］就会非常可取。

$$[H][H]^{\mathrm{T}}=[I] \tag{G.4}$$

其中［I］是一个单位矩阵。

根据二值对称原理，这就促成了 ICT 的提出［I6］。ICT 保持了相对幅值这样的结构，符号，二值对称性，以及变换系数元素中的正交性。

G.3 简单的 2 维 16 阶 ICT

16 阶 ICT 从 DCT-II［I7］矩阵中发展出来：

$$D(i,j)=\begin{cases}\frac{1}{4}, & i=1\\ \frac{\sqrt{2}}{4}\cos\left[\frac{(i-1)\left(j-\frac{1}{2}\right)\pi}{16}\right], & 1\leqslant j\leqslant 16\end{cases} \tag{G.5}$$

矩阵［D］$_{16}$可以用二值对称原理改进为［T］$_{16}$，如图 G.2 所示。那些点表示矩阵沿着里面绘制的实线按照奇偶交错的对称规律扩展到右侧。

15 个元素（x_0，x_1，…）被选为保持［T］$_{16}$正交性的整数。这种选择可被认为是实现复杂度、变换编码增益和避免效率之间的一种平衡［I20］。由 Cham and Fong［I20］提出的［T］$_{16}$是用于 AVS-video 或 H.264/AVC 的 8 阶 ICT 的一个扩展版本，它取决于要被集成进去的标准。因此在 AVS-video 和 H.264/AVC 中的实现就会生成两个不同的［T］$_{16}$。8 阶 ICT 的流图如图 G.3 和 G.4［I19］所示。它们可用基本的加法和移位运算实现（图 G.5）。

这里，元素 x_0，x_1，…，x_7表示预测残差或像素强度值，X_0，X_1，…，X_7表示变换系数。方程 G.6 到 G.17 给出图 G.3 和 G.4 的稀疏矩阵因数。

$$[A_8(\mathrm{H.264})]=[A_4][A_3][A_2][A_1] \tag{G.6}$$

$$
T_{16}=\left(\begin{array}{cccccccc|ccc}
x_0 & x_0 & x_0 & x_0 & x_0 & x_0 & x_0 & x_0 & x_0 & x_0 & \cdots \\
x_1 & x_3 & x_5 & x_7 & x_9 & x_{11} & x_{13} & x_{15} & -x_{15} & -x_{13} & \cdots \\
x_2 & x_6 & x_{10} & x_{14} & -x_{14} & -x_{10} & -x_6 & -x_2 & -x_2 & -x_6 & \cdots \\
x_3 & x_9 & x_{15} & -x_{11} & -x_5 & -x_1 & -x_7 & -x_{13} & x_{13} & x_7 & \cdots \\
x_4 & x_{12} & -x_{12} & -x_4 & -x_4 & -x_{12} & x_{12} & x_4 & x_4 & x_{12} & \cdots \\
x_5 & x_{15} & -x_7 & -x_3 & -x_{13} & x_9 & x_1 & x_{11} & -x_{11} & -x_1 & \cdots \\
x_6 & -x_{14} & -x_2 & -x_{10} & x_{10} & x_2 & x_{14} & -x_6 & -x_6 & x_{14} & \cdots \\
x_7 & -x_{18} & -x_3 & x_{15} & x_1 & x_{13} & -x_5 & -x_9 & x_9 & x_5 & \cdots \\
x_8 & -x_8 & -x_8 & x_8 & x_8 & -x_8 & -x_8 & x_8 & x_8 & -x_8 & \cdots \\
x_9 & -x_5 & -x_{13} & x_1 & -x_{15} & -x_3 & x_{11} & x_7 & -x_7 & -x_{11} & \cdots \\
x_{10} & -x_2 & x_{14} & x_6 & -x_6 & -x_{14} & x_2 & -x_{10} & -x_{10} & x_2 & \cdots \\
x_{11} & -x_1 & x_9 & x_{13} & -x_3 & x_7 & x_{15} & -x_5 & x_5 & -x_{15} & \cdots \\
x_{12} & -x_4 & x_4 & -x_{12} & -x_{12} & x_4 & -x_4 & x_{12} & x_{12} & -x_4 & \cdots \\
x_{13} & -x_7 & x_1 & -x_5 & x_{11} & x_{15} & -x_9 & x_3 & -x_3 & x_9 & \cdots \\
x_{14} & -x_{10} & x_6 & -x_2 & x_2 & -x_6 & x_{10} & -x_{14} & -x_{14} & x_{10} & \cdots \\
x_{15} & -x_{13} & x_{11} & -x_9 & x_7 & -x_5 & x_3 & -x_1 & x_1 & -x_3 & \cdots
\end{array}\right)
$$

图 G. 2　从 DCT-II 矩阵 $[D]_{16}$生成的矩阵 $[T]_{16}$（1991 IEEE）

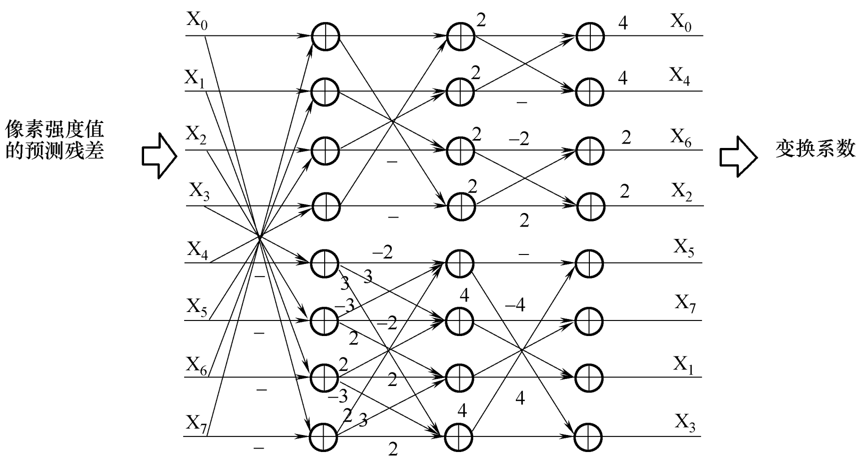


图 G. 3　AVS-video 中 8 点 ICT 正变换的流程图［I19］（2009 IEEE）

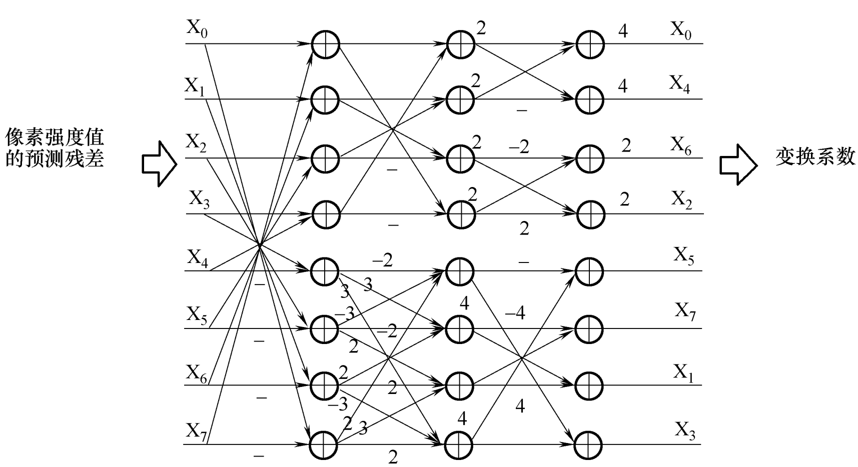


图 G. 4　H. 264/AVC 中 8 点 ICT 正变换的流程图［I19］（2009 IEEE）

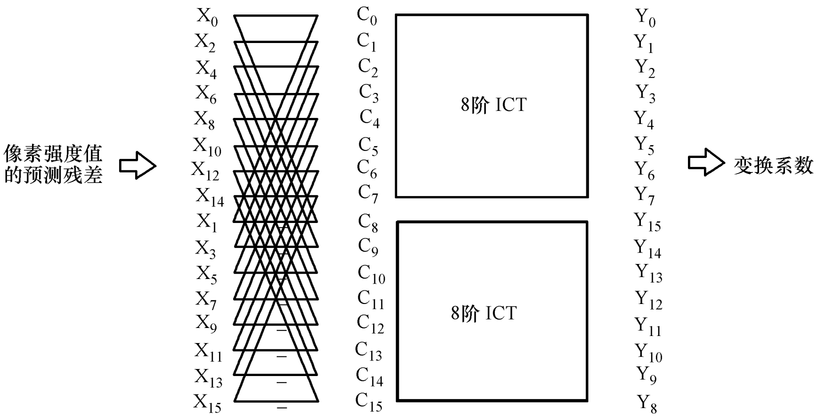


图 G.5　一维 16 点 ICT 正变换的流程图［I20］（2010 IEEE）

$$[A_1] = \begin{bmatrix} [I_4] & [I_4] \\ [I_4] & -[I_4] \end{bmatrix} \tag{G.7}$$

$$[A_2] = \mathrm{diag}\left[\begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & -1 & 0 \\ 1 & 0 & 0 & -1 \end{bmatrix}, \begin{bmatrix} -2 & -3 & 0 & 2 \\ 3 & -2 & 2 & 0 \\ 0 & 2 & 2 & 3 \\ 2 & 0 & -3 & 2 \end{bmatrix}\right] \tag{G.8}$$

$$[A_3] = \mathrm{diag}\left[\begin{bmatrix} 2 & 2 & 0 & 1 \\ 0 & -1 & 1 & 0 \\ 0 & 1 & -2 & 2 \\ 1 & 0 & 2 & 2 \end{bmatrix}, \begin{bmatrix} -1 & 0 & 0 & 4 \\ 0 & -4 & 1 & 0 \\ 0 & 1 & 4 & 0 \\ 4 & 0 & 0 & 1 \end{bmatrix}\right] \tag{G.9}$$

$$[A_4] = \mathrm{diag}[4[I_4], 2[I_2], [I_4]] \tag{G.10}$$

$$[A_{8(\mathrm{AVS})}] = [A_4][A_3][A_2][A_1] \tag{G.11}$$

$$[A_1] = \begin{bmatrix} [I_4] & [\hat{I}_4] \\ [\hat{I}_4] & -[I_4] \end{bmatrix} \tag{G.12}$$

$$[A_2] = \mathrm{diag}\left[\begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & -1 & 0 \\ 1 & 0 & 0 & -1 \end{bmatrix}, \begin{bmatrix} -2 & 0 & 0 & 3 \\ 0 & 2 & 3 & 0 \\ 0 & -5 & 2 & 0 \\ 3 & 0 & 0 & 2 \end{bmatrix}\right] \tag{G.13}$$

$$[A_3] = \mathrm{diag}\left[\begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 2 & 5 \\ 0 & 0 & -5 & 2 \end{bmatrix}, \begin{bmatrix} -2 & -3 & 0 & 2 \\ 3 & -2 & 2 & 0 \\ 0 & 2 & 2 & 3 \\ 2 & 0 & -3 & 2 \end{bmatrix}\right] \tag{G.14}$$

$$[A_4] = \mathrm{diag}[8[I_2],2[I_2],[I_4]] \tag{G. 15}$$

$$[I_k] = \begin{bmatrix} 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & & 0 & 0 \\ \vdots & & \ddots & & \vdots \\ 0 & 0 & & 1 & 0 \\ 0 & 0 & \cdots & 0 & 1 \end{bmatrix} \tag{G. 16}$$

$$(k \times k)$$

$$[\hat{I}_k] = \begin{bmatrix} 0 & 0 & \cdots & 0 & 1 \\ 0 & 0 & & 1 & 0 \\ \vdots & & \ddots & & \vdots \\ 0 & 1 & & 0 & 0 \\ 1 & 0 & \cdots & 0 & 0 \end{bmatrix} \tag{G. 17}$$

$$(k \times k)$$

$[T]_{16}$的流程图包括两个这样的并行实现的 8 点 ICT 正向变换块，如图 G. 5 所示［I20］。图 G. 5 的元素 X_0，X_1，……，X_{15}表示预测残差或像素亮度值，C_0，C_1，…，C_{15}是加法运算的中间结果，Y_0，Y_1，…，Y_{15}是生成的变换系数。

注意图 G. 6 和图 G. 7 中行的交替奇偶对称现象。图 G. 6 的稀疏矩阵因数是：

$$[M_{16(\mathrm{H.264})}] = [A_2][A_1] \tag{G. 18}$$

其中

$$[A_1] = \begin{bmatrix} [I_8] & [I_8] \\ [I_8] & -[I_8] \end{bmatrix} \tag{G. 19}$$

$$T_{16} = \begin{bmatrix}
8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 \\
12 & 12 & 10 & 10 & 6 & 6 & 3 & 3 & -3 & -3 & -6 & -6 & -10 & -10 & -12 & -12 \\
8 & 8 & 4 & 4 & -4 & -4 & -8 & -8 & -8 & -8 & -4 & -4 & 4 & 4 & 8 & 8 \\
10 & 10 & -3 & -3 & -12 & -12 & -6 & -6 & 6 & 6 & 12 & 12 & 3 & 3 & -10 & -10 \\
8 & 8 & -8 & -8 & -8 & -8 & 8 & 8 & 8 & 8 & -8 & -8 & -8 & -8 & 8 & 8 \\
6 & 6 & -12 & -12 & 3 & 3 & 10 & 10 & -10 & -10 & -3 & -3 & 12 & 12 & -6 & -6 \\
4 & 4 & -8 & -8 & 8 & 8 & -4 & -4 & -4 & -4 & 8 & 8 & -8 & -8 & 4 & 4 \\
3 & 3 & -6 & -6 & 10 & 10 & -12 & -12 & 12 & 12 & -10 & -10 & 6 & 6 & -3 & -3 \\
3 & -3 & -6 & 6 & 10 & -10 & -12 & 12 & 12 & -12 & -10 & 10 & 6 & -6 & -3 & 3 \\
4 & -4 & -8 & 8 & 8 & -8 & -4 & 4 & -4 & 4 & 8 & -8 & -8 & 8 & 4 & -4 \\
6 & -6 & -12 & 12 & 3 & -3 & 10 & -10 & -10 & 10 & -3 & 3 & 12 & -12 & -6 & 6 \\
8 & -8 & -8 & 8 & -8 & 8 & 8 & -8 & 8 & -8 & -8 & 8 & -8 & 8 & 8 & -8 \\
10 & -10 & -3 & 3 & -12 & 12 & -6 & 6 & 6 & -6 & 12 & -12 & 3 & -3 & -10 & 10 \\
8 & -8 & 4 & -4 & -4 & 4 & -8 & 8 & -8 & 8 & -4 & 4 & 4 & -4 & 8 & -8 \\
12 & -12 & 10 & -10 & 6 & -6 & 3 & -3 & -3 & 3 & -6 & 6 & -10 & 10 & -12 & 12 \\
8 & -8 & 8 & -8 & 8 & -8 & 8 & -8 & 8 & -8 & 8 & -8 & 8 & -8 & 8 & -8
\end{bmatrix}$$

图 G. 6　H. 264/AVC（SICT）的 16 阶变换矩阵［I20］（2010 IEEE）

表 G. 2　运算次数［I20］

简单 16 阶 ICT	移位次数	加法次数	总运算次数
	224	88	112

和

$$[A_2] = \mathrm{diag}[M_{8(\mathrm{H.264})},M_{8(\mathrm{H.264})}] \tag{G. 20}$$

$$
T_{16}=\begin{bmatrix}
8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8\\
10 & 10 & 9 & 9 & 6 & 6 & 2 & 2 & -2 & -2 & -6 & -6 & -9 & -9 & -10 & -10\\
10 & 10 & 4 & 4 & -4 & -4 & -10 & -10 & -10 & -10 & -4 & -4 & 4 & 4 & 10 & 10\\
9 & 9 & -2 & -2 & -10 & -10 & -6 & -6 & 6 & 6 & 10 & 10 & 2 & 2 & -9 & -9\\
8 & 8 & -8 & -8 & -8 & -8 & 8 & 8 & 8 & 8 & -8 & -8 & -8 & -8 & 8 & 8\\
6 & 6 & -10 & -10 & 2 & 2 & 9 & 9 & -9 & -9 & -2 & -2 & 10 & 10 & -6 & -6\\
4 & 4 & -10 & -10 & 10 & 10 & -4 & -4 & -4 & -4 & 10 & 10 & -10 & -10 & 4 & 4\\
2 & 2 & -6 & -6 & 9 & 9 & -10 & -10 & 10 & 10 & -9 & -9 & 6 & 6 & -2 & -2\\
2 & -2 & -6 & 6 & 9 & -9 & -10 & 10 & 10 & -10 & -9 & 9 & 6 & -2 & -2 & 2\\
4 & -4 & -10 & 10 & 10 & -10 & -10 & 10 & -10 & 10 & 10 & -10 & -10 & 10 & 4 & -4\\
6 & -6 & -10 & 10 & 2 & -2 & 9 & -9 & -9 & 9 & -2 & 2 & 10 & -10 & -6 & 6\\
8 & -8 & -8 & 8 & -8 & 8 & 8 & -8 & 8 & -8 & -8 & 8 & -8 & 8 & 8 & -8\\
9 & -9 & -2 & 2 & -10 & 10 & -6 & 6 & 6 & -6 & 10 & -10 & 2 & -2 & -9 & 9\\
10 & -10 & 4 & -4 & -4 & 4 & -10 & 10 & -10 & 10 & -4 & 4 & 4 & -4 & 10 & -10\\
10 & -10 & 9 & -9 & 6 & -6 & 2 & -2 & -2 & 2 & -6 & 6 & -9 & 9 & -10 & 10\\
8 & -8 & 8 & -8 & 8 & -8 & 8 & -8 & 8 & -8 & 8 & -8 & 8 & -8 & 8 & -8
\end{bmatrix}
$$

图 G.7　AVS-video（SICT）的 16 阶变换矩阵［I20］（2010 IEEE）

$$[M_{16(\mathrm{AVS})}]=[A_2][A_1] \qquad (\mathrm{G.21})$$

其中

$$[A_1]=\begin{bmatrix}[I_8] & [I_8]\\ [I_8] & -[I_8]\end{bmatrix} \qquad (\mathrm{G.22})$$

和

$$[A_2]=\mathrm{diag}[M_{8(\mathrm{AVS})},M_{8(\mathrm{AVS})}] \qquad (\mathrm{G.23})$$

可分离特性允许二维的 16×16 ICT 能用一维 16 阶 ICT 来实现（见图 G.1）。实现所需要的加法和移位次数见表 G.2。

G.4　改进的 2 维 16 阶 ICT

一般变换［T］$_{16}$的奇部（具有奇对称）矩阵元素的幅值相对高于偶部（具有偶对称）。这样，Dong 等人［I19］就提出了一种改进的 16 阶 ICT，其中的偶部保持不变，而奇部重新设计来降低 16 阶 DCT-II 的计算复杂度。

这样，就可以产生两个不同的 16 阶 ICT 分别用于 AVS-video 和 H.264/AVC 中的实现方案。图 G.8 给出了［T］$_{16}$的偶部［T］$_{8e}$、奇部［T］$_{8o}$以及改进的奇部［M］$_{8o}$。

［M］$_{8o}$的元素选择基于三点考虑：

- 元素幅值应该与偶部可以相比［T］$_{8e}$；
- 产生的阶次改进的 ICT（MICT）的第二基矢量应该类似于［T］$_{16}$；
- 可以开发出实现它的快速算法。

因为［M］$_{8o}$的元素不受正交性约束选择出来，他们可以具有比［M］$_{8o}$元素更小的幅值而不影响［T］$_{16}$固有的正交性。

产生的 MICT 矩阵（［M］$_{8o}$）的前三个基矢量类似于［T］$_{16}$。因此，低频系数保持不变，对于高清视频实现了很好的能量集中效果。

这样在高频端的系数中就可以实现计算复杂度和性能之间一种均衡。这使得将（x_1，x_3，x_5，x_7，x_9，x_{11}，x_{13}，x_{15}）选为（11，11，11，9，8，6，4，1）［I19］。

$$[T]_{8e}=\begin{bmatrix} x_0 & x_0 & x_0 & x_0 & x_0 & x_0 & x_0 & x_0 \\ x_2 & x_6 & x_{10} & x_{14} & -x_{14} & -x_{10} & -x_6 & -x_2 \\ x_4 & x_{12} & -x_{12} & -x_4 & -x_4 & -x_{12} & x_{12} & x_4 \\ x_6 & -x_{14} & -x_2 & -x_{10} & x_{10} & x_2 & x_{14} & -x_6 \\ x_8 & -x_8 & -x_8 & x_8 & x_8 & -x_8 & -x_8 & x_8 \\ x_{10} & -x_2 & x_{14} & x_6 & -x_6 & -x_{14} & x_2 & -x_{10} \\ x_{12} & -x_4 & x_4 & -x_{12} & -x_{12} & x_4 & -x_4 & x_{12} \\ x_{14} & -x_{10} & x_6 & -x_2 & x_2 & -x_6 & x_{10} & -x_{14} \end{bmatrix}$$

a)

$$[T]_{8o}=\begin{bmatrix} x_1 & x_3 & x_5 & x_7 & x_9 & x_{11} & x_{13} & x_{15} \\ x_3 & x_9 & x_{15} & -x_{11} & -x_5 & -x_1 & -x_7 & -x_{13} \\ x_5 & x_{15} & -x_7 & -x_3 & -x_{13} & x_9 & x_1 & x_{11} \\ x_7 & -x_{18} & -x_3 & x_{15} & x_1 & x_{13} & -x_5 & -x_9 \\ x_9 & -x_5 & -x_{13} & x_1 & -x_{15} & -x_3 & x_{11} & x_7 \\ x_{11} & -x_1 & x_9 & x_{13} & -x_3 & x_7 & x_{15} & -x_5 \\ x_{13} & -x_7 & x_1 & -x_5 & x_{11} & x_{15} & -x_9 & x_3 \\ x_{15} & -x_{13} & x_{11} & -x_9 & x_7 & -x_5 & x_3 & -x_1 \end{bmatrix}$$

b)

$$[M]_{8o}=\begin{bmatrix} x_1 & x_3 & x_5 & x_7 & x_9 & x_{11} & x_{13} & x_{15} \\ x_9 & x_{11} & x_{13} & x_{15} & -x_1 & -x_3 & -x_5 & -x_7 \\ x_5 & x_7 & -x_1 & -x_3 & -x_{13} & -x_{15} & x_9 & x_{11} \\ x_{15} & x_{13} & -x_{11} & -x_9 & x_7 & x_5 & -x_3 & -x_1 \\ x_{13} & -x_{15} & -x_9 & x_{11} & x_5 & -x_7 & -x_1 & x_3 \\ x_3 & -x_1 & -x_7 & x_5 & -x_{11} & x_9 & x_{15} & -x_{13} \\ x_7 & -x_5 & x_3 & -x_1 & -x_{15} & x_{13} & -x_{11} & x_9 \\ x_{11} & -x_9 & x_{15} & -x_{13} & x_3 & -x_1 & x_7 & -x_5 \end{bmatrix}$$

c)

图 G. 8 a) $[T]_{16}$的偶部 b) $[T]_{16}$的奇部 c) 改进的奇部 [I19] (2009 IEEE)

$[M]_{16}$可以用基本的加法和移位运算来实现（图 G. 9）。偶部是 $[T]_{8e}$，可以按照图 G. 3 和图 G. 4 来实现。$[M]_{8o}$可被分成三个矩阵 $[M]_1$、$[M]_2$和 $[M]_3$从而满足：

$$[M]_{8o}=[M]_1[M]_2[M]_3$$

矩阵 $[M]_1$、$[M]_2$和 $[M]_3$可以用加法和移位来表示，如 [I16] 解释那样。

注意到图 G. 10 和图 G. 11 中行的交替奇偶对称性。稀疏矩阵因数是：

$$[M_{16(\mathrm{H,264})}]=[A_2][A_1] \tag{G. 24}$$

其中

$$[A_1]=\begin{bmatrix} [I_8] & [I_8] \\ [I_8] & -[I_8] \end{bmatrix} \tag{G. 25}$$

且

$$[A_2]=\mathrm{diag}[M_{8(\mathrm{H.264})},M_{8o}] \tag{G. 26}$$

$$[M_{16(\mathrm{AVS})}]=[A_2][A_1] \tag{G. 27}$$

其中

$$[A_1]=\begin{bmatrix} [I_8] & [I_8] \\ [I_8] & -[I_8] \end{bmatrix} \tag{G. 28}$$

和

$$[A_2]=\mathrm{diag}[M_{8(\mathrm{AVS})},M_{8o}] \tag{G. 29}$$

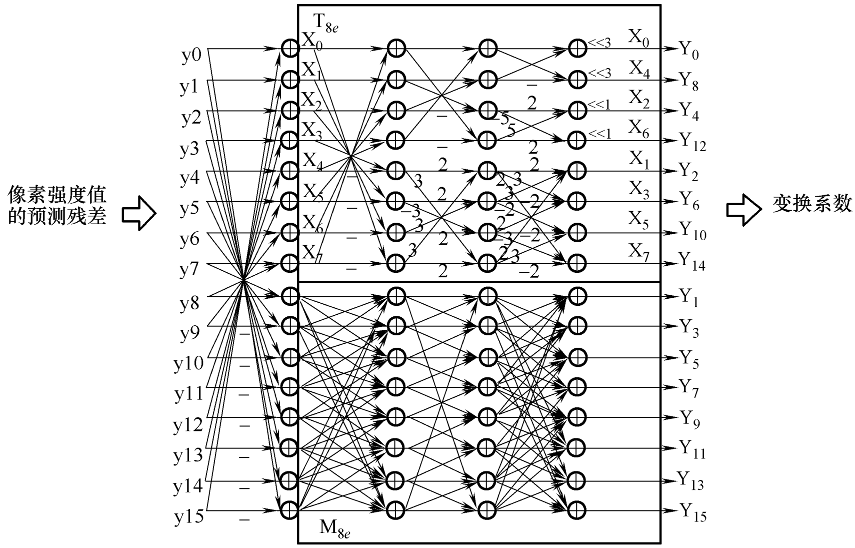


图 G.9 [M]$_{16}$的流程图（2007 IEEE）

$$
T_{16}=\left[\begin{array}{cccccccc|cccccccc}
8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 \\
11 & 11 & 11 & 9 & 8 & 6 & 4 & 1 & -1 & -4 & -6 & -8 & -9 & -11 & -11 & -11 \\
8 & 8 & 4 & 4 & -4 & -4 & -8 & -8 & -8 & -8 & -4 & -4 & 4 & 4 & 8 & 8 \\
8 & 6 & 4 & 1 & -11 & -11 & -11 & -9 & 9 & 11 & 11 & 11 & -1 & -4 & -6 & -8 \\
8 & 8 & -8 & -8 & -8 & -8 & 8 & 8 & 8 & 8 & -8 & -8 & -8 & -8 & 8 & 8 \\
11 & 9 & -11 & -11 & -4 & -1 & 9 & 8 & -8 & -9 & 1 & 4 & 11 & 11 & -9 & -11 \\
4 & 4 & -8 & -8 & 8 & 8 & -4 & -4 & -4 & -4 & 8 & 8 & -8 & -8 & 4 & 4 \\
1 & 4 & -6 & -8 & 9 & 11 & -11 & -11 & 11 & 11 & -11 & -9 & 8 & 6 & -4 & -1 \\
3 & -3 & -6 & 6 & 10 & -10 & -12 & 12 & 12 & -12 & -10 & 10 & 6 & -6 & -3 & 3 \\
4 & -1 & -8 & 6 & 11 & -9 & -11 & 11 & -11 & 11 & 9 & -11 & -6 & 8 & 1 & -4 \\
6 & -6 & -12 & 12 & 3 & -3 & 10 & -10 & -10 & 10 & -3 & 3 & 12 & -12 & -6 & 6 \\
11 & -11 & -9 & 11 & -6 & 8 & 1 & 4 & -4 & -1 & -8 & 6 & -11 & 9 & 11 & -11 \\
10 & -10 & -3 & 3 & -12 & 12 & -6 & 6 & 6 & -6 & 12 & -12 & 3 & -3 & -10 & 10 \\
9 & -11 & 11 & -11 & -1 & -4 & -6 & -8 & 8 & 6 & 4 & 1 & 11 & -11 & 11 & -9 \\
12 & -12 & 10 & -10 & 6 & -6 & 3 & -3 & -3 & 3 & -6 & 6 & -10 & 10 & -12 & 12 \\
6 & -8 & 1 & -4 & 11 & -11 & 9 & -11 & 11 & -9 & 11 & -11 & 4 & -1 & 8 & -6
\end{array}\right]
$$

图 G.10 面向 H.264/AVC（MICT）［I16］的 16 阶变换矩阵（2007 IEEE）

$$
[M_{8o}]=[M_1][M_2][M_3]=\begin{bmatrix}
-2 & 0 & 1 & -1 & -1 & 3 & -1 & 0 \\
3 & -1 & 1 & 1 & 0 & 2 & 0 & 1 \\
-1 & -3 & 1 & 0 & 1 & 0 & 2 & -1 \\
0 & 1 & 0 & 1 & 3 & 1 & -1 & -2 \\
1 & -1 & -3 & -2 & 0 & 1 & 0 & -1 \\
1 & 1 & 1 & 0 & -2 & 0 & 1 & -3 \\
0 & -2 & 0 & 1 & -1 & -1 & -3 & -1 \\
-1 & 0 & -2 & 3 & -1 & 1 & 1 & 0
\end{bmatrix}
$$

$$
\begin{bmatrix}
0 & 0 & 0 & 0 & 1 & -1 & 0 & 1 \\
0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\
0 & -1 & 0 & -1 & 0 & 0 & 1 & 0 \\
0 & -1 & 1 & 0 & 0 & 0 & -1 & 0 \\
0 & -1 & -1 & 1 & 0 & 0 & 0 & 0 \\
-1 & 0 & 0 & 0 & 0 & -1 & 0 & -1 \\
-1 & 0 & 0 & 0 & -1 & 0 & 0 & 1 \\
-1 & 0 & 0 & 0 & 1 & 1 & 0 & 0
\end{bmatrix}
$$

$$
\begin{bmatrix}
0 & -2 & 0 & -1 & 0 & 2 & -1 & 1 \\
0 & -1 & 0 & 2 & 1 & -1 & 0 & 2 \\
0 & -2 & 1 & 1 & 0 & 0 & 1 & -2 \\
-2 & 0 & -1 & 0 & 2 & 0 & -1 & -1 \\
-1 & 0 & 2 & 0 & -1 & -1 & -2 & 0 \\
-2 & 0 & 1 & -1 & 0 & 0 & 2 & 1 \\
-1 & 1 & 0 & 2 & -1 & 2 & 0 & 0 \\
-1 & -1 & -2 & 0 & -2 & -1 & 0 & 0
\end{bmatrix}
$$

$$
T_{16} = \left[\begin{array}{cccccccc|cccccccc}
8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 \\
11 & 11 & 11 & 9 & 8 & 6 & 4 & 1 & -1 & -4 & -6 & -8 & -9 & -11 & -11 & -11 \\
10 & 10 & 4 & 4 & -4 & -4 & -10 & -10 & -10 & 10 & -4 & -4 & 4 & 4 & 10 & 10 \\
8 & 6 & 4 & 1 & -11 & -11 & -11 & -9 & 9 & 11 & 11 & 11 & -1 & -4 & -6 & -8 \\
8 & 8 & -8 & -8 & -8 & -8 & 8 & 8 & 8 & 8 & -8 & -8 & -8 & -8 & 8 & 8 \\
11 & 9 & -11 & -11 & -4 & -1 & 9 & 8 & -8 & -9 & 1 & 4 & 11 & 11 & -9 & -11 \\
4 & 4 & -10 & -10 & 10 & 10 & -4 & -4 & -4 & -4 & 10 & 10 & -10 & -10 & 4 & 4 \\
1 & 4 & -6 & -8 & 9 & 11 & -11 & -11 & 11 & 11 & -11 & -9 & 8 & 6 & -4 & -1 \\
2 & -2 & -6 & 6 & 9 & -9 & -10 & 10 & 10 & -10 & -9 & 9 & 6 & -6 & -2 & 2 \\
4 & -1 & -8 & 6 & 11 & -9 & -11 & 11 & -11 & 11 & 9 & -11 & -6 & 8 & 1 & -4 \\
6 & -6 & -10 & 10 & 2 & -2 & 9 & -9 & -9 & 9 & -2 & 2 & 10 & -10 & -6 & 6 \\
11 & -11 & -9 & 11 & -6 & 8 & 1 & 4 & -4 & -1 & -8 & 6 & -11 & 9 & 11 & -11 \\
9 & -9 & -2 & 2 & -10 & 10 & -6 & 6 & 6 & -6 & 10 & -10 & 2 & -2 & -9 & 9 \\
9 & -11 & 11 & -11 & -1 & -4 & -6 & -8 & 8 & 6 & 4 & 1 & 11 & -11 & 11 & -9 \\
10 & -10 & 9 & -9 & 6 & -6 & 2 & -2 & -2 & 2 & -6 & 6 & -9 & 9 & -10 & 10 \\
6 & -8 & 1 & -4 & 11 & -11 & 9 & -11 & 11 & -9 & 11 & -11 & 4 & -1 & 8 & -6
\end{array}\right]
$$

图 G.11　面向 AVS-video（MICT）的 16 阶变换矩阵［I16］（2007 IEEE）

图 G.9 中，y_0，y_1 等是预测残差或像素强度值，Y_0，Y_1 等是变换系数。实现方案所需的加法和移位次数见表 G.3。

表 G.3　运算次数［I19］　2009　IEEE

改进的 16 阶 ICT	移位次数	加法次数	总的运算次数
	32	150	182

G.5　基于 Loeffler 因式分解的二维 16 阶 binDCT

DCT-II 可被分解为平面旋转和蝶形运算（图 G.12a）［I12，I8 和 I9］。就实现

算法所需的运算次数而言，Loeffler 等人［I5］（图 G.12）提出的 16 阶 DCT-Ⅱ的因式分解比 Chen 等人［I2］提出的更为有效。这种表示需要 31 次移位和 81 次加法［I5］。为了减少移位次数，平面旋转可以表示为三次剪切（原文：shear——译者注），或提升步骤［I10，I11］（图 G.12）。这样，理论上需要 4 次乘法和 2 次加法的旋转运算就可用 3 次乘法和 3 次加法来实现。

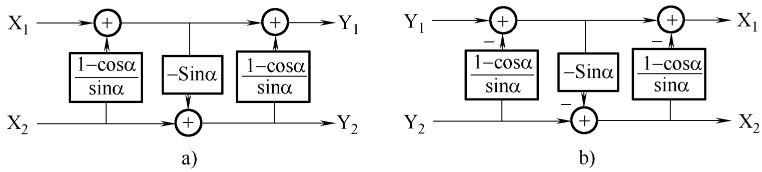


图 G.12 a）三个提升步骤表示的平面旋转 b）逆变换［I10，I11］

X_1，X_2 和 Y_1，Y_2（如图 G.12 a 和 b 所示）之间的关系由方程（G.30）~（G.35）导出。在许多情况下，平面旋转可进一步减少为 2 次乘法和 2 次加法。在图 G.14 中，p，u，K_1 和 K_2 表示乘法。如果在流程图中上一阶段存在平面旋转，那么 K_1 和 K_2 可被移入到缩放矩阵。剪切仍然包括一些参数，这些参数通常是无理数，但可以通过二值有理系数近似，如 Liang 和 Tran 提出的那样［I12］。

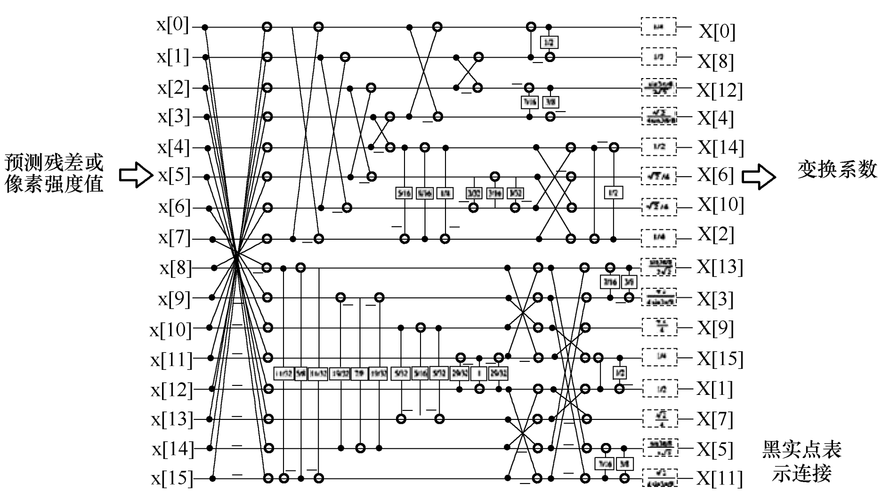


图 G.13 Loeffler 等提出 16 阶 DCT-Ⅱ流程图［I5］$\left(C_n=\cos\frac{n\pi}{32},\ S_n=\sin\frac{n\pi}{32}\right)$（1989 IEEE）

编码效率与计算复杂度之间的平衡通过调节近似表达式来实现。

$$Y_1 = X_1 + 2aX_2 + a^2bX_2 + abX_1 \tag{G.30}$$

$$Y_2 = X_2 + b(aX_2 + X_1) \tag{G.31}$$

其中

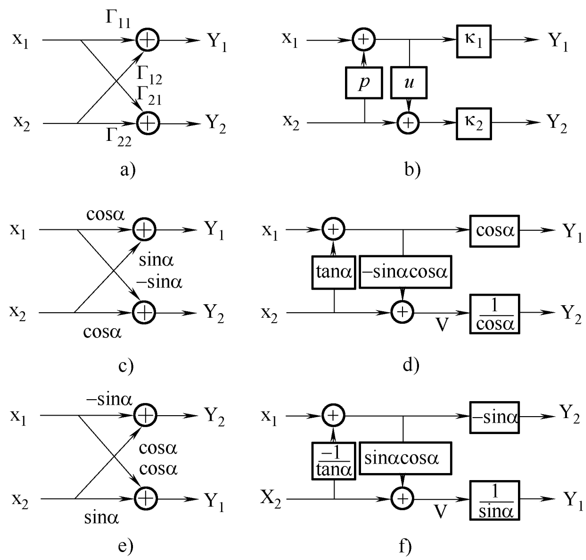


图 G. 14 a）一般的旋转 b）图 a 的缩放提升结构 c）正交的平面旋转 d）图 c 的缩放提升结构 e）图 c 的重排版本 f）图 c 的缩放提升结构 ［I12，I8，I9］（2001 IEEE）

$$a = \frac{1 - \cos\alpha}{\sin\alpha}, b = \sin\alpha \tag{G.32}$$

$$X_1 = Y_1 + 2\alpha Y_2 + a^2 bY_2 + abY_1 \tag{G.33}$$

$$X_2 = Y_2 + b(aY_2 + Y_1) \tag{G.34}$$

其中

$$a = -\frac{1 - \cos\alpha}{\sin\alpha}, b = -\sin\alpha \tag{G.35}$$

Liang 与 Tran 提出的基于 Loeffler 等人［I5］的 binDCT［I12］因式分解（binDCT-L）可以用最优次数的移位和乘法运算来实现 16 阶 DCT-II。尽管 binDCT 不完全是一个整数余弦变换，但它只用定点算术运算来计算。binDCT 的流程图如图 G. 15 所示。虚线框表示缩放因数。

16 阶 binDCT 的平移和加法次数由表 G. 4 给出。

表 G. 4 运算次数［I12］（2001 IEEE）

16 阶 binDCT-L	平移次数	加法次数	运算总次数
	51	106	157

G. 7 和 G. 8 节将讨论，以上 H. 264/AVC 和 AVS-video 的 16 阶 ICT 的实现细节，和基于变换编码增益和在各种分辨率视频序列中观察到的客观质量（BD-

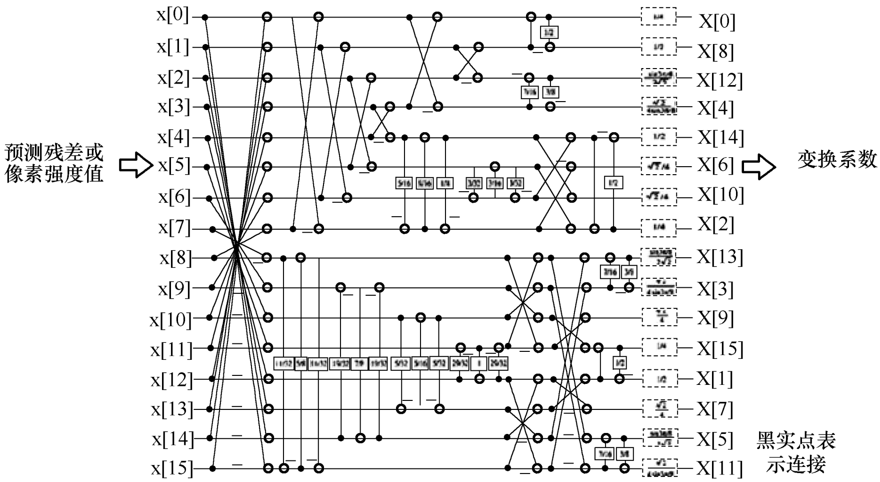


图 G. 15　16 阶 binDCT-L 的流程图（2001 IEEE）

PSNR 和 BD-比特率［U5］）的比较。

G. 6　变换编码增益

变换编码增益［B1］G_{TC}是一种变换的能量集中效果的测度。

$$G_{TC} = \frac{\frac{1}{N}\sum_{i=1}^{N}\sigma_{y_{ii}}^{2}}{\left(\prod_{i=1}^{N}\sigma_{y_{ii}}^{2}\right)^{1/N}} \qquad (G. 36)$$

式中，$\sigma_{y_{ii}}^{2}$是源信号元素的协方差。分子表示变换系数方差的算术平均。分母表示相应的几何平均。为了比较以前解释过的三种 ICT 的变换编码增益，我们把 1 维零均值单位方差的一阶马尔可夫过程（如图 G. 16 所示）作为信源。信源邻近元素的相关度可用ρ来表示。我们可以从表 G. 1 中看到，高清视频序列的ρ从 0. 5 到 0. 9 之间变化。图 G. 17 和表 G. 5 给出了对于所有三种 ICT 关于ρ的变换编码增益（以分贝计）变化及与 16 阶 DCT-II 比较的情况。这里，简单 16 阶 ICT（SICT）1 和 SICT 2 分别表示 H. 264/AVC 和 AVS-video 所实现的 ICT。改进的 16 阶 ICT（MICT）1 和 MICT 2 分别表示 H. 264/AVC 和 AVS-video 实现的 ICT。从图中我们可以得出结论，与 16 阶 DCT-II 相比（9. 4554 dB 对于$\rho = 0.95$），16 阶 binDCT-L 对于较高的ρ值（9. 4499 dB 对于$\rho = 0.95$）给出最好的变换编码增益。对于$\rho = 0.95$，改进的 16 阶 ICT 1 和 MICT2 分别给出 8. 8925dB 和 8. 8201dB。对于 SICT 1 和 SICT2 相应的值是 8. 7693dB 和 8. 7416dB（如图 G. 17 所示）。

ICT 的性能可以基于 BD-比特率和 BD-PSNR［U5］进行评价，这两项指标在（H. 264/AVC 和 AVS-video）与 ICT 集成的时候可以求出。这些值按照标准实现的

$$[T]=\begin{bmatrix}1 & \rho & \cdots & \rho^{14} & \rho^{15}\\ \rho & 1 & \ddots & \rho^{13} & \rho^{14}\\ \vdots & \ddots & \ddots & \rho & \vdots\\ \rho^{14} & \rho^{13} & \rho & 1 & \rho\\ \rho^{15} & \rho^{14} & \cdots & \rho & 1\end{bmatrix}$$

图 G. 16

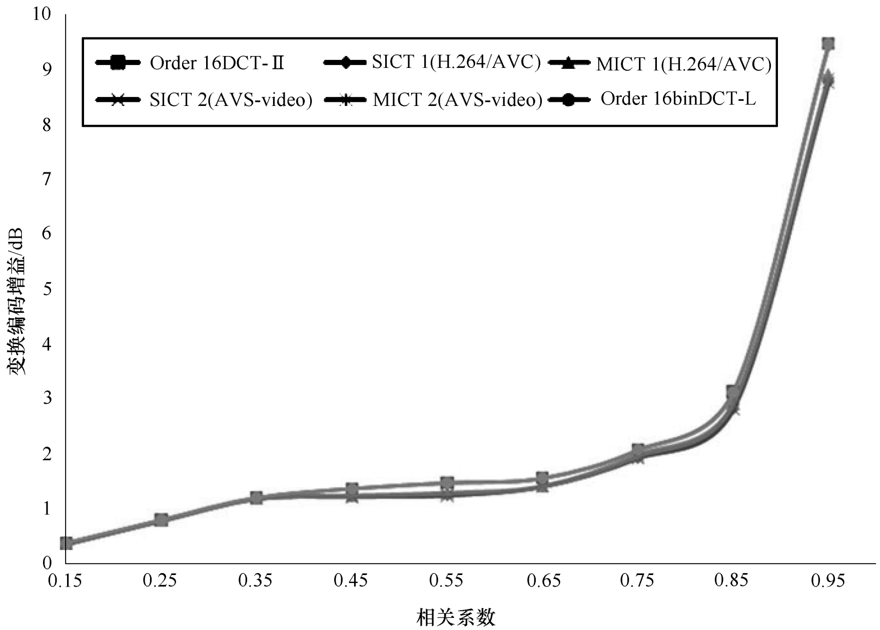


图 G. 17 各种 16 阶 ICT 和 16 阶 DCT-Ⅱ 的变换编码增益的比较［I21］

默认性能来计算。这里，BD-PSNR 给出相同比特率下的绝对 PSNR 增益，BD-比特率给出相同 PSNR 下比特率节约的百分比。

表 G. 5 变换编码增益关于 ρ 的变化［I21］

ρ	Order16 DCT-Ⅱ	SICT1 (H. 264/AVC)	MICT1 (H. 264/AVC)	SICT2 (AVS China)	MICT1 (AVS China)	16 阶 binDCT-L
0. 15	0. 3523	0. 3380	0. 3392	0. 3279	0. 3386	0. 3520
0. 25	0. 7715	0. 7618	0. 7654	0. 7609	0. 7632	0. 7713
0. 35	1. 1803	1. 1629	1. 1670	1. 1611	1. 1637	1. 1795
0. 45	1. 3496	1. 2078	1. 2184	1. 1955	1. 2169	1. 3480
0. 55	1. 4539	1. 2305	1. 2671	1. 2193	1. 2587	1. 4508
0. 65	1. 5383	1. 3863	1. 3887	1. 3842	1. 3906	1. 5362
0. 75	2. 0537	1. 9107	1. 9652	1. 8993	1. 9348	2. 0516
0. 85	3. 1185	2. 8030	2. 9170	2. 7951	2. 8879	3. 1024
0. 95	9. 4554	8. 7693	8. 8925	8. 7416	8. 8201	9. 4499

G.7 H.264/AVC 中的实现方案与性能分析

H.264/AVC 参考软件［H30］JM 18.0 中实现了三种 2-D 16 阶 ICT。用于简单和改进的 2 维 16 阶 ICT 的缩放和量化矩阵与参考软件中的一样，由于它们是 8 阶 ICT 的扩展版本；而 2 维 16 阶 binDCT-L 的相应矩阵则在附录中规定。变换大小的选择基于宏块级的 R-D 代价［G2］。H.264 高档次用于编码视频序列。这些视频序列可以从［V5］下载。表 G.6 规定了一些用于编码的配置参数。编码的执行利用了一个具有 Intel i7 四核的 2.6GHz 处理器支持 6GB 内存的系统。所用的操作系统是 Windows 7。属于高清（1920×1080，1280×720），WVGA（832×480）和 WQVGA（416×240）的视频序列用于评价性能。编码结果［每个序列的比特率对 PSNR，比特率节约百分比（BD-比特率）对 BD-PSNR］列在表格之中。还给出了每种不同分辨率序列的曲线图（见图 G.18～图 G.20 和表 G.7～表 G.9）。

表 G.6 配置参数［I21］

帧组大小	8
帧组结构	IBBBBBBP
帧内编码帧周期	0.5s
率失真优化	On
QP（量化参数）	22，27，32，37
参考帧	2
快速运动估计	On
搜索范围	32
去块效应滤波器	On
熵编码	CABAC

表 G.7 BD-比特率和 BD-PSNR（SICT 实现的 H.264/AVC）［**I21**］

序列名	BD-比特率（%）	BD-PSNR/dB
Vidyo1	−2.57	0.19
Vidyo2	−3.81	0.22
Vidyo3	−1.91	0.12

表 G.8 BD-比特率和 BD-PSNR（MICT 实现的 H.264/AVC）［**I21**］

序列名	BD-比特率（%）	BD-PSNR/dB
Vidyo1	−5.30	0.31
Vidyo2	−3.55	0.26
Vidyo3	−2.08	0.20

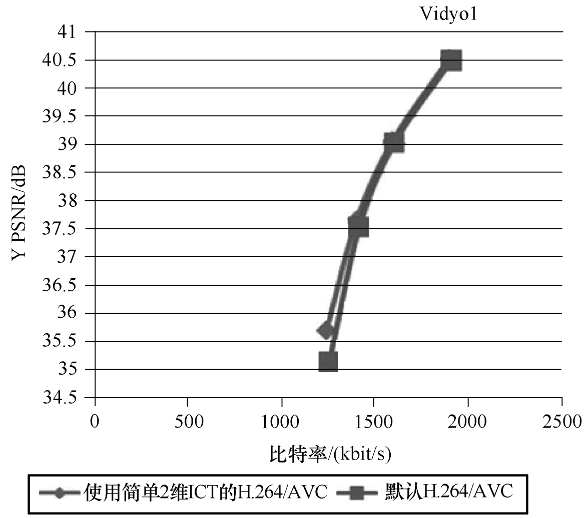


图 G. 18　Vidyol 序列 Y- PSNR 关于比特率的变化曲线（使用 SICT 的 H. 264/AVC）

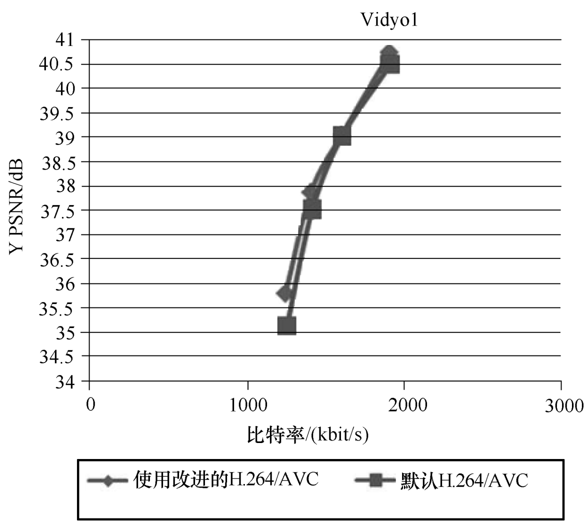


图 G. 19　Vidyol 序列的 Y- PSNR 关于比特率的变化曲线（使用 MICT 的 H. 264/AVC）［I21］

表 G. 9　BD- 比特率和 BD- PSNR（binDCT- L 实现的 H. 264/AVC）［**I21**］

序列名	BD- 比特率（%）	BD- PSNR/dB
Vidyo1	－4. 73	0. 36
Vidyo2	－4. 35	0. 31
Vidyo3	－1. 59	0. 19

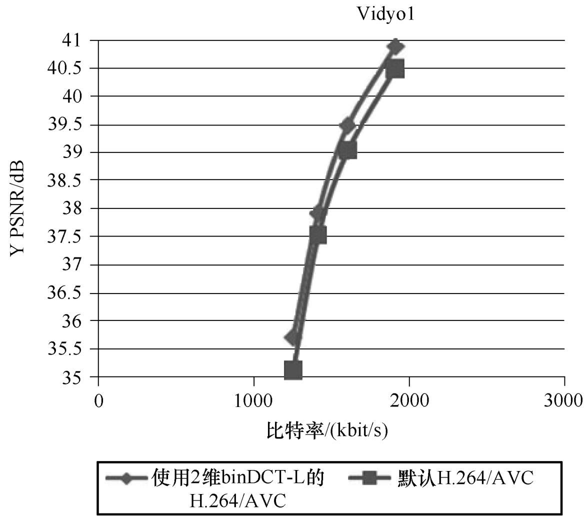


图 G. 20 Vidyol 序列的 Y-PSNR 关于比特率的变化曲线（使用 binDCT-L 的 H. 264/AVC）［I21］

G. 8 AVS-Video 的实现方案和性能分析

AVS-video 的参考软件［AS3］RM 52e 实现了三种 2 维 16 阶 ICT。简单和改进的 2 维 16 阶 ICT 的缩放和量化矩阵与参考软件 RM 52e 中的一样，由于它们是 8 阶 ICT 的扩展版本；而 2 维 16 阶 binDCT-L 的相应矩阵则在附录中规定。

表 G. 10 配置参数［I21］

帧组大小	8
帧组结构	IBBBBBBP
帧内编码帧周期	0. 5s
率失真优化	On
QP（量化参数）	22，27，32，37
参考帧	2
快速运动估计	On
搜索范围	32
去块效应滤波器	On
熵编码	CABAC

变换大小的选择基于宏块级的率失真代价［U5］。AVS 加强档次用于视频序列编码。表 G. 10 规定了一些用于编码的配置参数。编码的执行利用了一个具有 Intel i7 四

核的 2.6GHz 处理器支持 6GB 内存的系统。所用的操作系统是 Windows 7。属于高清（1920×1080，1280×720），WVGA（832×480）和 WQVGA（416×240）的视频序列用于评价性能。编码结果［每个序列的比特率对 PSNR，比特率节约百分比（BD-比特率）对 BD-PSNR］列在表格之中。还给出了来自每种不同分辨率序列的曲线图（见图 G.21～图 G.23 和表 G.11～表 G.13）。

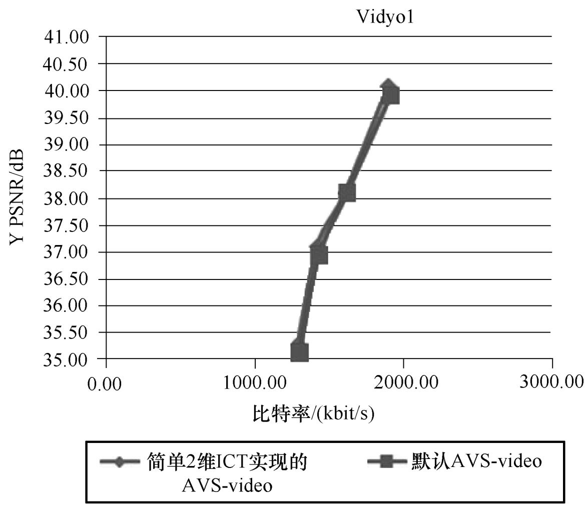


图 G.21 Vidyol 序列的 Y-PSNR 关于比特率的变化曲线（SICT 实现的 AVS-video）［I21］

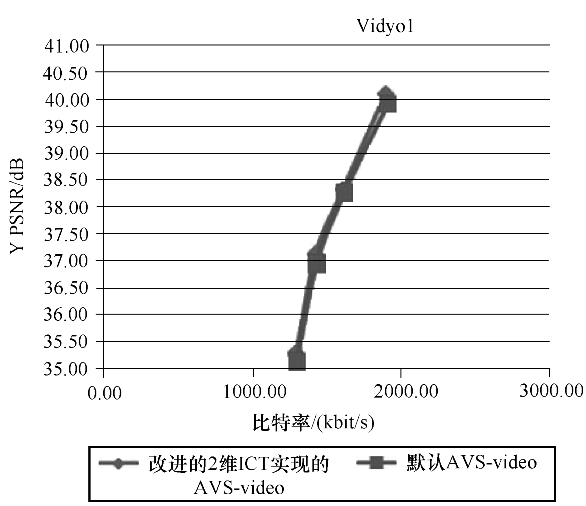


图 G.22 Vidyo 1 序列的 Y-PSNR 关于比特率的变化曲线（MICT 实现的 AVS-video）［I21］

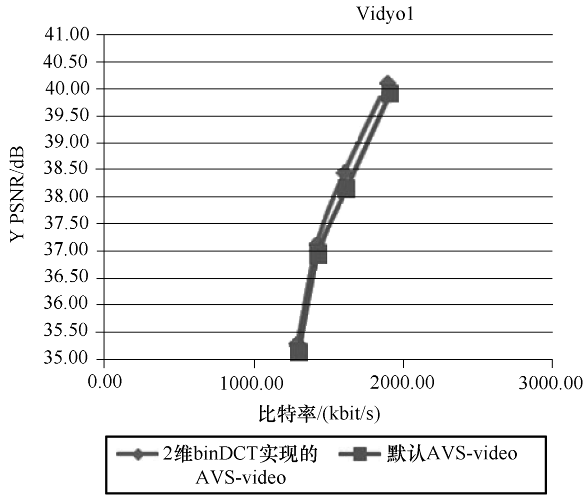


图 G. 23 Vidyo 1 序列的 Y PSNR 关于比特率的变化曲线（binDCT-L 实现的 AVS-video）[I21]

表 G. 11 BD-比特率和 BD-PSNR（SICT 实现的 AVS-video）**[I21]**

序列名称	BD-比特率（%）	BD-PSNR/dB
Vidyo1	-5. 18	0. 29
Vidyo2	-6. 57	0. 34
Vidyo3	-3. 23	0. 21

表 G. 12 BD-比特率和 BD-PSNR（MICT 实现的 AVS-video）**[I21]**

序列名称	BD-比特率（%）	BD-PSNR/dB
Vidyo1	-2. 57	0. 34
Vidyo2	-2. 68	0. 26
Vidyo3	-3. 12	0. 33

表 G. 13 BD-比特率和 BD-PSNR（binDCT-L 实现的 AVS-video）**[I21]**

序列名称	BD-比特率（%）	BD-PSNR/dB
Vidyo1	-7. 45	0. 41
Vidyo2	-5. 28	0. 26
Vidyo3	-2. 14	0. 13

G. 9 结论和未来的工作

H. 264/AVC 实现的 SICT、MICT 和 binDCT-L 的 BD-PSNR 和 BD-比特率表明，

与标准中目前的默认 ICT（2-D 8×8 and 2-D 4×4 ICT）相比，对于高清视频（1920×1280，1280×720）可以实现相当大的比特率节约和 PSNR 增益。幅值的 BD-比特率或正值的 BD-PSNR 表示比较好的编码效率。而且，对于较低分辨率视频（WQVGA，WVGA）SICT、MICT 和 binDCT-L 的编码效率类似于默认的 ICT。binDCT-L 在三个被实现的 2 维 16 阶 ICT 中给出最好的编码效率。这是由于 binDCT-L 在全部变换中是 2 维 16 阶 DCT-II 的最佳近似，而且可用相对少的平移和加法就能实现。从 G.7 节和 G.8 节的图中我们可以推断出，AVS-video 实现的 SICT、MICT 和 binDCT-L 的 BD-PSNR 和 BD-比特率遵循类似于 H.264/AVC 的曲线规律。唯一的差别在于，对于相同的比特率 AVS-video 给出相对较低的 PSNR 增益；此外对于较低分辨率的视频 SICT、MICT 和 binDCT-L 的性能尽管与默认 ICT（2 维 8×8）可相比拟，但仍次于 H.264/AVC。binDCT 在此也给出最佳性能。

［I21］的研究工作只涉及 2 维 16 阶 ICT。我们可以开发更高阶（32×32，64×64）的 2 维 ICT 并类似地运用于各标准之中。这些更高阶的 ICT 对于超高分辨率视频（4096×2560）能给出较好的编码效率，但是它们会增加计算复杂度。视频编码联合协作组（JCT-VC，Joint Collaboration Team on Video Coding）正在考虑将 2 维 32 阶 ICT 和 2 维 64 阶 ICT 集成到高效视频编码标准（HEVC）［E5］（H.264/AVC 的继承标准）中。正在考虑中的测试模型 TmuC 0.9 是 HEVC 测试模型的最新版本。

参 考 文 献

AVS China: Video/Systems

A1. GB/T 20090.1, Information technology—Advanced coding of audio and video—Part 1: system, Chinese AVS standard (2009)

A2. China National Standard: GB/T 200090.2-2006, Information technology—Advanced coding of audio and video—Part 2: video (2006)

A3. AVS Video Experts Group, Information technology—Advanced coding of audio and video—Part 2: Video (AVS1-P2 JQP FCD 1.0), Audio Video Coding Standard Group of China (AVS), Doc. AVS-N1538 (2008)

A4. AVS Video Experts Group, Information technology—Advanced coding of audio and video—Part 3: Audio, Audio Video Coding Standard Group of China (AVS), Doc. AVS-N1551 Sept 2008 (Final, March 2009)

A5. D. Wu et al., An exp-Golomb encoder and decoder architecture for JVT/AVS, in *Proceeding of 5th International Conference on ASIC*, vol. 2, Beijing, China, pp. 910–913 Oct 2003

A6. W. Gao et al., *AVS—The Chinese Next-Generation Video Coding Standard* (NAB, Las Vegas, 2004)

A7. Q. Wang et al., Context-based 2D-VLC for video coding, in *IEEE International Conference on Multimedia and Expo (ICME)*, vol. 1, pp. 89–92, June 2004

A8. R. Wang et al., Sub-pixel motion compensation interpolation filter in AVS video coding, in *IEEE International Conference on Multimedia and Expo (ICME)*, vol. 1, Taipei, Taiwan, pp. 93–96, June 2004

A9. Z. Nan et al., Spatial prediction based intra-coding, in *IEEE International Conference on Multimedia and Expo, ICME,* vol. 1, Taipei, Taiwan, pp. 97–100, June 2004

A10. L. Fan, S.W. Ma, F. Wu, Overview of AVS video standard, in *IEEE International Conference on Multimedia and Expo, ICME*, vol. 1, Taipei, Taiwan, pp. 423–426, June 2004

A11. B. Sheng, W. Gao and D. Wu, A platform-based architecture of loop filter for AVS, in *IEEE 7th International Conference on Signal Processing, ICSP*, vol. 1, Beijing, China, pp. 571–574, Aug–Sept 2004

A12. C. Peng et al., Architecture of AVS hardware decoding system, in *IEEE Proceedings of the International Symposium on Intelligent Multimedia, Video and Speech Process, ISIMP'04*, Hong Kong, China, pp. 306–309, Oct 2004

A13. W. Gao, AVS standard—Audio video coding standard workgroup of China, in *International Conference on Wireless and Optical Communications, 14th Annual WOCC* 2005, p. 54, 22–23 April 2005

A14. L. Yu et al., Overview of AVS-video: tools, performance and complexity, in *SPIE VCIP*, vol. 5960, Beijing, China, pp. 596021-1–596021-12, July 2005
A15. H. Zhao, G. Wang, G. Hou, AVS on satellite. *SPIE VCIP 2005*, vol. 5960, Beijing, China, pp. 596023-1–596023-6, July 2005
A16. H. Qi et al., A study on the motion vector prediction schemes for AVS. *SPIE VCIP 2005*, vol. 5960, Beijing, China, pp. 596066-1–596066-8, July 2005
A17. B. Sheng, W. Gao, D. Wu, An implemented VLSI architecture of inverse quantizer for AVS HDTV video decoder, in *IEEE 6th International Conference ASIC*, vol. 1, Shanghai, China, pp. 244–247, Oct 2005
A18. AVS project and AVS-video techniques. Power-point slides by L. Yu, Chair of AVS video. http://www-uta.edu/dip/Courses/EE5351/ISPACSAVS.pdf. Dec 2005
A19. L. Miao et al., Context-dependent bit plane coding in China AVS audio, in *Proceedings of 2005 International Symposium on Intelligent Signal Processing and Communication Systems, ISPACS 2005*, pp. 765–768, 13–16 Dec 2005
A20. J. Dong, K.N. Ngan, 16 $\times$ 16 Integer cosine transform for HD video coding, in *PCM 2006*, Springer, LNCS 4261, pp. 114–121 (2006)
A21. B. Sheng et al., An efficient VLSI architecture of VLD for AVS HDTV decoder. IEEE Trans. Consum. Electron. **52**, 696–701 (2006)
A22. X.-F. Wang, D.-B. Zhao, Performance comparison of AVS and H.264/AVC video coding standards. J. Comput. Sci. Technol. **21**(3), 310–314 (May 2006)
A23. Q. Wang et al., Context-based 2D-VLC entropy coder in AVS video coding standard. J. Comput. Sci. Technol. **21**(3), 315–322 (2006). (ACM digital library)
A24. F. Yi et al., Low-complexity tools in AVS Part 7. J. Comput. Sci. Technol. **21**(3), 345–353 (2006). (Special section on China AVS standard, ACM digital library)
A25. S.-W. Ma, W. Gao, Low complexity integer transform and adaptive quantization optimization. J. Comput. Sci. Technol. **21**(3), 354–359 (2006)
A26. L. Zhang, D. Xie, D. Wu, Improved FFSBM algorithm and its VLSI architecture for AVS video standard. J. Comput. Sci. Technol. **21**(3), 378–382 (May 2006)
A27. Z. Ma et al., Intra coding of AVS Part 7 video coding standard. Technical Report, May 2006. https://sites.google.com/site/visionzhanma/publications
A28. B. Sheng, W. Gao, D. Xie, Algorithmic and architectural co-design for integer motion estimation of AVS. IEEE Trans. Consum. Electron. **52**, 1092–1098 (2006)
A29. S. Yao et al., A hardware implementation for full-search motion estimation of AVS with search center prediction. IEEE Trans. Consum. Electron. **52**, 1356–1361 (2006)
A30. Z. Zhang et al., A flexible VLSI architecture of transport processor for an AVS HDTV decoder SoC. IEEE Trans. Consum. Electron. **52**, 1427–1432 (2006)
A31. H. Jia et al., An AVS HDTV video decoder architecture employing efficient HW/SW partitioning. IEEE Trans. Consum. Electron. **52**, 1447–1453 (Nov. 2006)
A32. L. Yu et al., Entropy coding based on code-value compact memory structure for AVS video coding standard, in *IEEE 8th International Conference on Signal Processing*, vol. 2, Guilin, China, pp. 16–20, Nov 2006
A33. K. Zhang, L. Yu, An area-efficient VLSI architecture for AVS intra frame encoder, in *SPIE VCIP 2007*, vol. 6508, San Jose, CA, pp. 650822-1–650822-10, 29 Jan 2007
A34. D. Zhou, P. Liu, A hardware-efficient dual-standard VLSI architecture for MC interpolation in AVS and H.264, in *IEEE ISCAS 2007*, Seattle, WA, pp. 2910–2913, May 2007
A35. K. Zhang, X.-Y. Wu, L. Yu, An area-efficient VLSI implementation of CA-2D-VLC decoder for AVS, in *IEEE ISCAS 2007*, Seattle, WA, pp. 3151–3154, May 2007

A36. Y. Qu, Y. Li, S. Mei, A cost-effective VLSI architecture of VLD for MPEG-2 and AVS, in *IEEE International Conference on Multimedia and Expo 2007*, Beijing, China, pp. 1619–1622, 2–5 July 2007

A37. Q. Zhang, Y. Fang, C. Wang, A novel rate control algorithm for AVS video coding, in *IEEE International Conference on Wireless Communication, Networking and Mobile Computing, WiCom 2007*, Shanghai, China, pp. 2900–2902, Sept 2007

A38. Z. Yang et al., DSP implementation of deblocking filter for AVS, in *IEEE ICIP*, vol. 6, San Antonio, TX, pp. 205–208, Sept–Oct 2007

A39. K. Luo et al., A parallel serial filtering mixed advanced ID interpolation architecture for AVS, in *IEEE 7th International Conference on ASIC*, Guilin, China, pp. 762–765, Oct 2007

A40. W. Gao, T. Huang, AVS Standard—status and future plan, in Workshop on multimedia new technologies and application, Shenzhen, China, Oct 2007

A41. Y.H. Lu, Adaptive search range based fast quarter-pixel motion estimation for AVS, in *MIPPR 2007: Remote Sensing and GIS Data Processing and Applications, and Innovative Multispectral Technology and Applications*, *Proceeding of SPIE*, vol. 6790, Wuhan, China, pp. 67904D-1–67904D-8, Nov 2007

A42. M.-L. Wong, Y.-L. Lin, H.H. Chen, A hardware-oriented intra prediction scheme for high definition AVS encoder. PCS, Lisbon, Portugal, Nov 2007

A43. L. Deng, X.D. Xie, W. Gao, A real-time full architecture for AVS motion estimation. IEEE Trans. Consum. Electron. **53**, 1744–1751 (2007)

A44. X. Ji et al., B-picture coding in AVS video compression standard, in *Signal Processing: Image Communication*, vol. 23, pp. 31–41, Jan 2008 (The constraint of 2 reference buffers for both P- and B-pictures)

A45. D. Ding, S. Yao, L. Yu, Memory bandwidth efficient hardware architecture for AVS encoder. IEEE Trans. Consum. Electron. **54**, 675–680 (May 2008)

A46. H.B. Yin et al., .An efficient VLSI architecture for rate distortion optimization in AVS video encoder, in *IEEE ISCAS*, Seattle, WA, pp. 2805–2808, May 2008

A47. S. Hu, X. Zhang, Z. Yang, Efficient implementation of interpolation for AVS, in *IEEE Congress on Image and Signal Processing, CISP'08*, vol. 3, Sanya, China, pp. 133–138, May 2008

A48. Q. Yang et al., An efficient hardware implementation for intra prediction of AVS encoder, in *IEEE ICALIP 2008*, Shanghai, China, pp. 200–205, July 2008

A49. L. Wang et al., Hardware implementation of transform and quantization for AVS encoder, in *IEEE ICALIP 2008*, Shanghai, China, pp. 843–847, July 2008

A50. L. Xu et al., Hardware architecture for AVS entropy encoder. IEEE Trans. Consum. Electron. **54**, 769–777 (May 2008)

A51. X. Jin, S. Li, K.N. Ngan, AVS video standard implementation for SoC design, in *IEEE International Conference on Neural Networks and Signal Processing*, Zhenjiang, China, pp. 660–665, June 2008

A52. T. Zhang, W. Wang, J. He, On the pre-echo control method in transient signal coding of AVS audio, in *IEEE International Conference on Audio, Language and Image Processing, ICALIP 2008*, Shanghai, China, pp. 242–246, July 2008

A53. L. Xu, L. Deng, X. Ji, X. Peng, W. Gao, Hardware architecture of AVS entropy encoder. IEEE Trans. Consum. Electron. **54**, 769–777 (May 2008)

A54. C. Guanghua et al., An efficient VLSI architecture of sub-pixel interpolator for AVS encoder, in *9th International Conference on Signal Processing 2008, ICSP 2008*, pp. 1255–1258, 26–29 Oct 2008

A55. G.-A. Su et al., Low-cost hardware-sharing architecture of fast 1-D inverse transforms for H.264/AVC and AVS applications. IEEE Trans. Circuits Syst II Express Briefs **55**(12), 1249–1253 (2008)

A56. B.Y. Lei, K.T. Lo, H. Lei, Design and optimization of MC and DF in AVS-M decoder, in *IEEE ICCEE, International Conference on Computer and Electrical Engineering*, Hong Kong, China, pp. 922–925, Dec 2008
A57. B. Tang et al., AVS encoder performance and complexity analysis based on mobile video communication, in *WRI International Conference on Communications and Mobile Computing, CMC '09*, vol. 3, pp. 102–107, 6–8 Jan 2009
A58. W. Gao, K.N. Ngan, L. Yu, Special issue on AVS and its application. Signal Process. Image Commun. **24**, 245–344 (2009)
A59. L. Yu, S. Chen, J. Wang, Overview of AVS-video coding standards. Signal Process. Image Commun. **24**, 247–262 (2009)
A60. L. Zhang et al., Context-based entropy coding in AVS video coding standard. Signal Process. Image Commun. **24**, 263–276 (2009)
A61. Z.M. Wu, Y. He, Combined adaptive-fixed interpolation with multi-directional filters. Signal Process. Image Commun. **24**, 277–286 (2009)
A62. D. Ding et al., Reconfigurable video coding framework and decoder reconfiguration instantiation of AVS. Signal Process. Image Commun. **24**, 287–299 (2009)
A63. Z.J. Yang, W.K. Wan, X.M. Chen, AVS trick modes for PVR and VOD services. Signal Process. Image Commun. **24**, 300–311 (2009)
A64. X. Jin, S. Li, K.N. Ngan, Platform-independent MB-based AVS video standard implementation. Signal Process. Image Commun. **24**, 312–323 (April 2009)
A65. J. Zheng et al., An efficient VLSI architecture for CBAC of AVS HDTV decoder. Signal Process. Image Commun. **24**, 324–332 (April 2009)
A66. Y.G. Wang et al., Robust dual watermarking algorithm for AVS video. Signal Process. Image Commun. **24**, 333–344 (April 2009)
A67. S. Zhang, J.P. Li, An improved CMMB FEC scheme based on BI-STBC, in *IEEE CISP 2009*
A68. H.B. Yin et al., Hardware friendly mode decision algorithm for high definition AVS video encoder, in *IEEE CISP 2009*
A69. C.H. Shi, Y.H. Shi, B.C. Yin, Feedback-based error tracking for AVS-M, in *IEEE CISP 2009*
A70. S. Sreedhar, K.R. Rao, Multiplexing and demultiplexing of AVS China video with AAC audio, in *TELSIKS 2011*, Nis, Serbia, 5–8 Oct 2011
A71. P.R. Ramolia, Low complexity AVS-M using machine learning algorithm C4.5, in *TELSIKS 2011*, Nis, Serbia, 5–8 Oct 2011
A72. D. Sahana, K.R. Rao, A study on AVS-M standard. Advanced Computational Technologies published by Romanian Academy Publishing House (2011)
A73. AVS Work Group website: http://www.avs.org.cn/en/
A74. Information technology—Advanced coding of audio and video—Part 7: Mobile video
A75. Y. Shen et al., A simplified intra prediction method. AVS Doc. AVS-M 1419 (2004)
A76. F. Yi et al., An improvement of intra prediction mode coding. AVS Doc. AVS-M 1456 (2004)
A77. L. Xiong, Improvement of chroma intra prediction. AVS Doc. AVS-M1379 (2004)
A78. X. Mao et al., Adaptive block size coding for AVS-X profile. AVS Doc. AVS-M2372 (2008)
A79. GB 20090.2 RTP Payload Format, FG IPTV- 0512, International Telecommunication Union, May 2007
A80. C. Bin, *The Introduction of AVS Part-1: System* (EE Dept, UTA, 2012)
A81. L. Fan, *AVS-M: Mobile video standard in Mobile Multimedia Broadcasting standards*, (Springer, 2009), pp. 485–508
A82. S. Ma, S. Wang, W. Gao, *Overview of 1857 video coding standard*, IEEE ICIP (2013)

Advanced Image Coding (AIC)

AC1. AIC website: http://www.bilsen.com/aic/
AC2. J.J. Hwang, S.G. Cho, Proposal for objective distortion metrics for AIC standardization. ISO/IEC JTC 1/SC 29/WG 1 N4548 Mar 2008
AC3. Z. Zhang, R. Veerla, K.R. Rao, A modified advanced image coding, in *IEEE Proceedings of CANS 2008*, Targu Mures, Romania, pp. 110–116, Nov 8–10 2008

AVS China Software

AS1. Website of AVS working group: http://www.avs.org.cn/en
AS2. M. Clendenin, China's AVS codec gains more top-tier support. http://www.eetimes.com/news/latest/showArticle.jhtml?articleID=199900715. 4 June 2007
AS3. Link for AVS reference software (RM 52e): ftp://159.226.42.57/public/avs_doc/avs_software (Requires password)
AS4. http://bmrc.berkeley.edu/research/mpeg/media/pattern.gif

Books

B1. N.S. Jayant, P. Noll, *Digital Coding of Waveforms: Principles and Applications to Speech and Video* (Prentice-Hall, Englewood Cliffs, 1984)
B2. K.R. Rao, P. Yip, *Discrete Cosine Transform—Algorithms, Advantages, Applications* (Academic Press, Boca Raton, 1990)
B3. K.R. Rao, J.J. Hwang, *Techniques and Standards for Image, Video and Audio Coding*. (Prentice Hall, 1996)
B4. J.D. Murray, W. van Ryper, *Encyclopedia of Graphics File Formats*, 2nd edn. (O'Reilly Media 1996), online available: http://www.fileformat.info/mirror/egff/index.htm
B5. W.K. Pratt, *Digital Image Processing* (Wiley, New York, 2001)
B6. A. Papoulis, S.U. Pillai, *Probability, Random Variables, and Stochastic Processes* (McGraw-Hill, Boston, 2002)
B7. K.R. Rao, Z.S. Bojkovic, D.A. Milovanovic, Multimedia Communication Systems—Techniques, Standards, and Networks. (Prentice Hall PTR, 2002)
B8. I. Richardson, *H.264 and MPEG-4 Video Compression—Video Coding for Next-Generation Multimedia.* (Wiley, 2003)
B9. M. Ghanbari, *Standard Codecs: Image Compression to Advanced Video Coding* (IEE, London, 2003)
B10. L. Harte, Introduction to MPEG: MPEG-1, MPEG-2 and MPEG-4. (Althos, 2006)
B11. H.R. Wu, K.R. Rao (eds.), *Digital Video Image Quality and Perceptual Coding*. (Taylor and Francis, CRC Press, 2006)
B12. K. Sayood, *Introduction to Data Compression*, 3rd edn. (Morgan Kaufmann Publishers, 2006)
B13. Z. Wang, A.C. Bovik, *Modern Image Quality Assessment*. (Morgan and Claypool Publishers, 2006)
B14. J.-B. Lee, H. Kalva, *The VC-1 and H.264 Video Compression Standards for Broadband Video Services.* (Springer, New York, 2008)
B15. W. Fischer, *Digital Video and Audio Broadcasting Technology—A Practical Engineering Guide*, 2nd edn. (Springer, 2008)
B16. B. Furht, S.A. Ahson, *Handbook of mobile broadcasting, DVB-H, DMB, ISDB-T and MEDIAFLO* (CRC Press, Boca Raton, FL, 2008). (H.264 related chapters)

B17. F.-L. Luo, *Mobile multimedia broadcasting standards*. (Springer, 2009)
B18. I.E. Richardson, *H.264 Advanced Video Compression Standard*, 2nd edn. (Wiley, Hoboken, 2010)
B19. K.R. Rao, D.N. Kim, J.J. Hwang, *Fast Fourier Transform—Algorithms and Applications*. (Springer, 2010)
B20. C.T. Chen, *Signals and Systems*, 3rd edn. (Oxford University Press, New York, 2004)

SMPTE Video Coder 1 (VC 1)

C1. Microsoft, WMV-9—an advanced video codec for 3GPP. 3GPP SA4 Meeting #18, document S4-030613, Sept 2003. Available from http://www.3gpp.org/ftp/tsg_sa/WG4_CODEC/TSGS4_28/Docs/
C2. S. Srinivasan et al., Windows media video 9: overview and applications. Signal Process. Image Commun. **19**, 851–875 (2004)
C3. S. Srinivasan, S.L. Regunathan, An overview of VC-1, in *Proceedings of SPIE/VCIP*, vol. 5960, Beijing, China, pp. 720–728, July 2005
C4. A. Klemets, *RTP Payload Format for Video Codec 1 (VC-1)*. (The Internet Society, 2006)
C5. J.-B. Lee, H. Kalva, An efficient algorithm for VC-1 to H.264 video transcoding in progressive compression, in *IEEE International Conference on Multimedia and Expo (ICME)*, Toronto, Canada, pp. 53–56, July 2006
C6. VC-2 Video Compression, SMPTE ST 2042-1:2012
C7. VC-1 Compressed Video Bitstream Format and Decoding Process. SMPTE 421M-2006, SMPTE Standard (2006)
C8. M. Pantoja, N. Ling, W. Shang, Coefficient conversion for transform domain VC-1 to H.264 transcoding. IEEE Workshop on Signal Processing Systems, pp. 363–367, 2007
C9. M. Tammen, M. El-Sharkawy, M. Rizkalla, Fast block-size selection using residual texture for VC-1, in *National Radio Science Conference*, pp. 1–7, 2007
C10. M. Pantoja, H. Kalva, J.-B. Lee, P-frame transcoding in VC-1 to H.264 transcoders, in *IEEE ICIP*, vol. 5, San Antonio, TX, pp. 297–300, Sept–Oct 2007
C11. H. Kalva, J.-B. Lee, The VC-1 and H.264 video compression standards, in *IEEE Multimedia*, vol. 14, pp. 88–91, Oct–Dec 2007
C12. SMPTE, VC1 bitstream transport encodings, RP 227–2007
C13. R. Citro et al., A multi-standard micro-programmable deblocking filter architecture and its application to VC-1 video decoder, in *IEEE International SOC Conference*, pp. 225–228, 2008
C14. Y.-H. Lim, J.-S. Kang, An efficient architecture of bitplane coding with high frame rate for VC-1. Signal Process. Image Commun. **23**, 692–698 (Oct. 2008)
C15. Y.L. Lee, T.Q. Nguyen, Analysis and efficient architecture design for VC-1 overlap smoothing and in-loop deblocking filter, in *IEEE Trans. CSVT*, vol. 18, pp. 1786–1796, Dec 2008
C16. J. Loomis, M. Wasson, *VC-1 Technical Overview*. (Microsoft Corporation, Oct 2007). http://www.microsoft.com/windows/windowsmedia/howto/articles/vc1techoverview.aspx
C17. Microsoft Windows Media: http://www.microsoft.com/windows/windowsmedia
C18. C.-P. Fan, G.-A. Su, Efficient fast 1-D 8 × 8 inverse integer transform for VC-1 application. IEEE Trans. CSVT **19**(4), 584–590 (2009)
C19. H. Sliman et al., All-zero block detection in VC-1, in *IEEE International Midwest Symposium on Circuits and Systems*, pp. 555–560, 2009
C20. R. Citro et al., Programmable deblocking filter architecture for a VC-1 video decoder. IEEE Trans. CSVT **19**(8), 1227–1233 (2009)

C21. C.-P. Fan, G.-A. Su, Fast algorithm and low-cost hardware-sharing design of multiple integer transforms for VC-1. IEEE Trans. Circuits Syst. II **56**(10), 788–792 (2009)
C22. Y.-J. Wang et al. Low-complexity integrated architecture of 4 × 4, 4 × 8, 8 × 4 and 8 × 8 inverse integer transforms of VC-1, in *IEEE Internaional Midwest Symposium on Circuits and Systems*, pp. 543–546, 2009
C23. M. Pantoja, N. Ling, Low complexity rate control for VC-1 to H.264 transcoding, in *IEEE International Symposium on Circuits and Systems*, Taipei, Taiwan, pp. 888–891, May 2009
C24. M. Pantoja, N. Ling, Adaptive transform size and frame-field selection for efficient VC-1 to H.264 high profile transcoding, in *IEEE International Symposium on Circuits and Systems,* Taipei, Taiwan, pp. 2357–2360, May 2009
C25. H.A. Deghiedy, M. El-Sharkawy, G. Fahmy, VC-1 zero block detection method in motion estimation video coding. *IEEE International Symposium on Signal Processing and Information Technology*, pp. 246–249, 2010
C26. M.J. Tammen et al., Advanced motion estimation techniques for VC-1, in *IEEE International Symposium on Signal Processing and Information Technology*, pp. 240–245, 2010
C27. J.-Y. Lee, J.-J. Lee, S.M. Park, Multi-core platform for an efficient H.264 and VC-1 video decoding based on macro block row-level parallelism. IET Circuits, Devices Syst. **4**(2), 147–158 (2010)
C28. Y.-C. Chao et al., An efficient architecture of multiple 8 × 8 transforms for H.264/AVC and VC-1 decoders, in *International Conference on Green Circuits and Systems*, pp. 595–598, 2010
C29. V. Vijayakumar, Low complexity H.264 to VC-1 transcoder. M.S. Thesis, EE Dept, University of Texas at Arlington, 2010
C30. K.R. Rao, D.N. Kim, Current video coding standards: H.264/AVC, Dirac, AVS China and VC-1. IEEE Southeastern Symposium on System Theory (SSST), pp. 1–8, Mar 2010
C31. SMPTE website: http://www.smpte.org/home/
C32. Microsoft website: http://www.microsoft.com/en/us/default.aspx

Dirac Video Codec

D1. T. Borer, T. Davies, Dirac—Video compression using open technology, in *EBU Technical Review*, pp. 1–9, July 2005. Available: http://www.ebu.ch/en/technical/trev/trev_home.html
D2. H. Eeckhaut et al., Speeding up Dirac's entropy coder, in *Proceedings of the 5th WSEAS International Conference on Multimedia, Internet and Video Technologies*, Corfu, Greece, pp. 120–125, Aug 2005
D3. T. Davies, A modified rate-distortion optimization strategy for hybrid wavelet video coding, in *IEEE ICASSP*, vol. 2, Toulouse, France, pp. 909–912, May 2006
D4. M. Tun, W.A.C. Fernando, An error-resilient algorithm based on partitioning of the wavelet transform coefficients for a Dirac video codec, in *IEEE Tenth International Conference on Information Visualization*, London, England, pp. 615–620, July 2006
D5. K. Onthriar, K.K. Loo, Z. Xue, Performance comparison of emerging Dirac video codec with H.264/AVC, in *IEEE International Conference on Digital Telecommunication, ICDT'06*, Cap Esterel, Cote d'Azur, France, pp. 22–22, Aug 2006
D6. M. Tun, K.K. Loo, J. Cosmas, A novel rate control algorithm for the Dirac video codec based upon the quality factor optimization, in *IEEE International Conference on Geometric Modeling and Imaging, GMAI'07*, Zürich, Switzerland, pp. 14–18, July 2007
D7. M. Tun, K.K. Loo, J. Cosmas, Error-resilient performance of Dirac video codec over packet-erasure channel. IEEE Trans. Broadcast. **53**, 649–659 (2007)

D8. M. Tun, K.K. Loo, J. Cosmas, Semi-hierarchical motion estimation for the Dirac video codec, in *IEEE International Symposium on Broadband Multimedia Systems and Broadcasting*, *BMS '08*, Las Vegas, NV, pp. 1–6, Mar–Apr 2008

D9. Dirac Video, Dirac specification [Online] (2008). Available: http://diracvideo.org/download/specification/dirac-spec-latest.pdf

D10. M. Tun, K.K. Loo and J. Cosmas, "Rate control algorithm based on quality factor optimization for Dirac video codec," *Signal Processing: Image Commun.*, Elsevier B.V., vol. 23, pp. 649–664, Oct. 2008.

D11. T. Davies, The Dirac algorithm (2008), http://dirac.sourceforge.net/documentation/algorithm/

D12. T. Borer, Dirac coding: tutorial and implementation. EBU Networked Media Exchange Seminar, June 2009

D13. C. Gargour et al., A short introduction to wavelets and their applications. IEEE Circuits Syst. Mag. **9**, 57–68 (2009). (II Quarter)

D14. A. Ravi, Performance analysis of the Dirac video codec. M.S. Thesis, EE Dept., UTA, 2009

D15. A. Ravi, K.R. Rao, Performance analysis and comparison of the Dirac video codec with H.264 / MPEG-4 Part 10 AVC. IJWMIP **4**, 635–654 (2011)

D16. A. Urs, Multiplexing/demultiplexing Dirac video and AAC audio while maintaining lip sync. M.S. Thesis, EE Dept., UTA, 2011

D17. A. Urs, K.R. Rao, Multiplexing/de-multiplexing Dirac video with AAC audio bit stream. TELSIKS 2011, Nis, Serbia, 5–8 Oct 2011

D18. Daubechies wavelet: http://en.wikipedia.org/wiki/Daubechies_wavelet

D19. Daubechies wavelet filter design: http://cnx.org/content/m11159/latest/

D20. Dirac software and source code: http://diracvideo.org/download/dirac-research/

D21. Dirac video codec—A programmer's guide: http://dirac.sourceforge.net/documentation/code/programmers_guide/toc.htm

D22. The Dirac web page and developer support: http://dirac.sourceforge.net

D23. Dirac Pro: http://www.bbc.co.uk/rd/projects/dirac/diracpro.shtml

D24. DiracPRO software: http://dirac.kw.bbc.co.uk/download/

D25. BBC Research on Dirac: http://www.bbc.co.uk/rd/projects/dirac/index.shtml

D26. BBC Research, The technology behind Dirac [Online]. Available: http://www.bbc.co.uk/rd/projects/dirac/technology.shtml

D27. C. Gargour et al., A short introduction to wavelets and their applications. IEEE Circuits Syst. Mag. **9**, 57–68 (2009). (II Quarter)

D28. H. Eeckhaut et al., Tuning the M-coder to improve Dirac's Entropy Coding, http://escher.elis.ugent.be/publ/Edocs/DOC/P105_088.pdf

D29. K. Noam, B. Tamir, Dirac video codec: optimizing software performance using architectural considerations. Technion—Israel Institute of Technology, Electrical Engineering Faculty, Software lab Performance Tuning

D30. CMPT 365 Course Slides, School of Computing Science, Simon Fraser University, Fig. 3: http://www.cs.sfu.ca/CourseCentral/365/li/material/notes/Chap4/Chap4.3/Chap4.3.html

D31. Digital down converter: http://en.wikipedia.org/wiki/Digital_down_converter

D32. Streaming SIMD extensions (SSE): http://msdn.microsoft.com/en-us/library/t467de55%28VS.71%29.aspx

D33. Intel Compilers: http://software.intel.com/en-us/intel-compilers/

D34. Microsoft Visual Studio add-ins: http://en.wikipedia.org/wiki/List_of_Microsoft_Visual_Studio_add-ins

D35. GStreamer: http://www.gstreamer.net/

Data Compression

DC1. G.G. Langdon Jr, An introduction to arithmetic coding. IBM J. Res. Dev. **28**, 135–149 (1984)

DC2. T.A. Welch, A technique for high-performance data compression. Computer **17**(6), 8–19 (1984)

DC3. I.H. Witten, R.M. Neal, J.G. Cleary, Arithmetic coding for data compression. Commun. ACM **30**, 520–540 (1987)

DC4. F. Wheeler, Adaptive arithmetic coding source code, http://www.fredwheeler.org/ac/

DC5. M. Burrows, D.J. Wheeler, A block-sorting lossless data compression algorithm. Technical Report 124, Digital Equipment Corporation, 1994 (Open source bzip2 uses the Burrows-Wheeler block-sorting transformation (BWT) and Huffman coding)

HEVC (High Efficiency Video Coding)/NGVC (Next Generation Video Coding)

E1. K.H. Lee et al., Technical considerations for Ad Hoc Group on new challenges in video coding standardization. ISO/IEC MPEG 85th meeting, M15580, Hanover, Germany, July 2008

E2. E. Alshina et al., Technical considerations on new challenges in video coding standardization. ISO/IEC MPEG 86th meeting, M15899, Busan, South Korea, Oct 2008

E3. Y. Ye, M. Karczewicz, Improved H.264 intra coding based on bi-directional intra prediction, directional transform, and adaptive coefficient scanning, in *IEEE International Conference Image Process'08 (ICIP08)*, San Diego, USA, pp. 2116–2119, Oct 2008

E4. G.J. Sullivan, The high efficiency video coding (HEVC) standardization initiative. Power Point slides 7 June 2010

E5. G.J. Sullivan, J.-R. Ohm, Recent developments in standardization of high efficiency video coding (HEVC), in *Proceedings of SPIE, Applications of Digital Image Processing* XXXIII, vol. 7798, San Diego, CA, pp. 77980V-1–V-7, 1–3 Aug 2010

E6. R. Joshi, Y.A. Reznik, M. Karczewicz, Efficient large size transforms for high-performance video coding, in *Proceedings of SPIE*, vol. 7798, San Diego, CA, Aug 2010

E7. IEEE Transaction on CSVT, vol. 20, Special section on HEVC (several papers), Dec 2010

E8. M. Karczewicz et al., A hybrid video coder based on extended macroblock sizes, improved interpolation and flexible motion representation. IEEE Trans. CSVT **20**, 1698–1708 (2010)

E9. T. Wiegand, B. Bross, W.-J. Han, J.-R. Ohm, G. J. Sullivan, WD3: working draft 3 of high-efficiency video coding, joint collaborative team emerging HEVC standard on video coding (JCT-VC) of ITU-T VQEG and ISO/IEC MPEG. Doc. JCTVC-E603, Geneva, CH, Mar 2011

E10. S. Jeong et al., High efficiency video coding for entertainment quality. ETRI J. **33**, 145–154 (2011)

E11. K. Asai et al., New video coding scheme optimized for high-resolution video sources. IEEE J. Sel. Top. Sign. Proces. **5**(7), 1290–1297 (2011)

E12. IEEE Journal of Selected Topics in Signal Processing, vol. 5(7) (2011) (several papers on HEVC), Introduction to the issue on emerging technologies for video compression

E13. Z. Ma, H. Hu, Y. Wang, On complexity modeling of H.264/AVC video decoding and its application for energy efficient decoding. IEEE Trans. Multimedia **13**(6), 1240–1255 (2011)

E14. J. Dong, K.N. Ngan, Adaptive pre-interpolation filter for high efficiency video coding. J. VCIR **22**, 697–703 (2011)
E15. W. Deng et al., Fast mode dependent directional transform via butterfly-style transform and integer lifting steps. J. VCIR **22**, 721–726 (2011)
E16. F. De Simone et al., Towards high efficiency video coding; subjective evaluation of potential coding technologies. J. VCIR **22**, 734–748 (2011)
E17. B. Bross et al., High efficiency video coding (HEVC) text specification draft 6. Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, 7th Meeting, Geneva, CH, 21–30 Nov 2011
E18. Z. Ma, A. Segall, Low resolution decoding for high-efficiency video coding. IASTED SIP-2011, Dallas, TX, Dec 2011
E19. Y-J. Yoon et al., Adaptive prediction block filter for video coding. ETRI J. **34**, 106–109 (2012), (See the references) http://etrij.etri.re.kr
E20. B. Li, G.J. Sullivan, J. Xu, Compression performance of high efficiency video coding (HEVC) working draft 4, in *IEEE ISCAS*, session B1L-H, Seoul, Korea, pp. 886–889, May 2012
E21. N. Ling, High efficiency video coding and its 3D extension: a research perspective. Keynote Speech, in *IEEE ICIEA*, pp. 2153–2158, Singapore, July 2012
E22. C. Fogg, Suggested figures for the HEVC specification. ITU-T & ISO/IEC JCTVC-J0292r1, July 2012
E23. J.-R, Ohm, G. Sullivan, T. Wiegand, Developments and trends in 2D and 3D video coding standardization: HEVC and more, in *IEEE ICIP 2012*, Tutorial, Orlando, FL, 30 Sept 2012
E24. A. Saxena, F.C. Fernandes, On secondary transforms for prediction residual, in *IEEE ICIP* 2012, Orlando, FL, pp. 2489–2492, Sept 2012
E25. High efficiency video coding, MA.P2, IEEE ICIP 2012, Poster session, (several papers), Orlando, FL, Sept–Oct 2012. These are listed below (E26–E38)
E26. S. Ma, J. Si, S. Wang, A study on the rate distortion modeling for high efficiency video coding, in *IEEE ICIP 2012*, Orlando, FL, pp. 181–184, Oct 2012
E27. A. Gabriellini et al., Spatial transform skip in the emerging high efficiency video coding standard, in *IEEE ICIP 2012*, Orlando, FL, pp. 185–188, Oct 2012
E28. H. Vianna et al., High performance hardware architectures for the inverse rotational transform of the emerging HEVC standard, in *IEEE ICIP 2012*, Orlando, FL, pp. 189–192, Oct 2012
E29. H. Aoki, Efficient quantization parameter coding based on intra/inter prediction for visual quality conscious video coders, in *IEEE ICIP 2012*, Orlando, FL, pp. 193-196, Oct 2012
E30. X. Zhang et al., New chroma intra prediction modes based on linear model for HEVC, in *IEEE ICIP 2012*, Orlando, FL, pp. 197–200, Oct 2012
E31. D. Palomino, A memory aware and multiplierless VLSI architecture for the complete intra prediction of the HEVC emerging standard, in *IEEE ICIP 2012*, Orlando, FL, pp. 201–204, Oct 2012
E32. H. Schwarz, Extension of high efficiency video coding (HEVC) for multiview video and depth data, in *IEEE ICIP 2012*, Orlando, FL, pp. 205-208, Oct 2012
E33. M. Budagavi, V. Sze, Unified forward+inverse transform architecture for HEVC, in *IEEE ICIP 2012*, Orlando, FL, pp. 209–212, Oct 2012
E34. C.C. Chi, Improving the parallelization efficiency of HEVC decoding, in *IEEE ICIP 2012*, Orlando, FL, pp. 213–216, Oct 2012
E35. G. Correa et al., Motion compensated tree depth limitation for complexity control of HEVC encoding, in *IEEE ICIP 2012*, Orlando, FL, pp. 217–220, Oct 2012
E36. M. Zhang, C. Zhao, J. Xu, An adaptive fast intra mode decision in HEVC, in *IEEE ICIP 2012*, Orlando, FL, pp. 221–224, Oct 2012

E37. J. Stankowski et al., Extensions of the HEVC technology for efficient multiview video coding, in *IEEE ICIP 2012*, Orlando, FL, pp. 225–228, Oct 2012

E38. L. Kerofsky, K. Misra, A. Segall, On transform dynamic range in high efficiency video coding, in *IEEE ICIP 2012*, Orlando, FL, pp. 229–232, Oct 2012

E39. M.E. Sinangil et al., Memory cost vs. coding efficiency trade-offs for HEVC motion estimation engine, in *IEEE ICIP 2012*, Orlando, FL, pp. 1533–1536, Sept–Oct 2012

E40. M.E. Sinangil et al., Hardware - aware motion estimation search algorithm development for HEVC standard, in *IEEE ICIP 2012*, Orlando, FL, pp. 1529–1532, Sept–Oct 2012

E41. M. Zhang, C. Zhao, J. Xu, An adaptive fast intra mode decision in HEVC, in *IEEE ICIP 2012*, Orlando, FL, pp. 221–224, Sept–Oct 2012

E42. H. Zhang, Z. Ma, Fast intra prediction for high efficiency video coding. Pacific Rim Conference on Multimedia, PCM2012, Singapore, Dec 2012

E43. Special issue on emerging research and standards in next generation video coding. IEEE Trans. CSVT **22** (2012)

E44. B. Shrestha, Investigation of image quality of Dirac, H.264 and H.265. Report for EE 5359: Multimedia Processing, EE Dept., UTA, Arlington, TX (2012)

E45. B. Li, G.J. Sullivan, J. Xu, Compression performance of high efficiency video coding working draft 4. IEEE ISCAS 2012, Session on HEVC, Seoul, Korea, pp. 886–889, May 2012

E46. http://www.h265.net has info on developments in HEVC NGVC—Next generation video coding

E47. JVT KTA reference software http://iphome.hhi.de/suehring/tml/download/KTA/

E48. JM11KTA2.3 (http://www.h265.net/2009/04/kta-software-jm11kta23.html)

E49. Permanent link: mode-dependent directional transform (MDDT) in JM/KTA TMUC HEVC Software—http://hevc.kw.bbc.co.uk/svn/jctvc-tmuc/

E50. F. Bossen, D. Flynn, K. Sühring, HEVC reference software manual, (July 2011) Online available:
http://phenix.int-evry.fr/jct/doc_end_user/documents/6_Torino/wg11/JCTVC-F634-v2.zip

E51.
http://phenix.it-sudparis.eu/jct/doc_end_user/documents/7_Geneva/wg11/JCTVC-G399-v3.zip

E52. http://www.itu.int/ITU-T/studygroups/com16/jct-vc/index.html

E53. JCT-VC documents are publicly available at http://ftp3.itu.ch/av-arch/jctvc-site and http://phenix.it-sudparis.eu/jct/

E54. HEVC Model 9.0, available for download at https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/tags/HM-9.0/. The latest version available at the time this chapter was prepared is HM-9.0. Version HM 8.0 is available at: http://hevc.kw.bbc.co.uk/trac/browser/tags/HM-8.0

E55. J. Nightingale, Q. Wang, C. Grecos, HEVStream: a framework for streaming and evaluation of high efficiency video coding (HEVC) content in loss-prone networks. IEEE Trans. Consum. Electron. **59**, 404–412 (2012)

E56. D. Marpe et al., Improved video compression technology and the emerging high efficiency video coding standard, in *IEEE International Conference on Consumer Electronics*, Berlin, Germany, pp. 52–56, Sept 2011

E57. HM9 High efficiency video coding (HEVC) Test Model 9 (HM 9) encoder description JCTVC-K1002v2, Shanghai meeting, Oct 2012

E58. B. Bross et al., High efficiency video coding (HEVC) text specification draft 8. JCTVC-J1003, July 2012. http://phenix.int-evry.fr/jct/doc_end_user/current_document.Php?id=5889

E59. G.J. Sullivan et al., Overview of the high efficiency video coding (HEVC) standard. IEEE Trans. CSVT **22**, 1649–1668 (2012)
E60. J.-R. Ohm et al., Comparison of the coding efficiency of video coding standards – including high efficiency video coding (HEVC). IEEE Trans. CSVT **22**, 1669–1684 (2012)
E61. F. Bossen et al., HEVC complexity and implementation analysis. IEEE Trans. CSVT **22**, 1685–1696 (2012)
E62. B. Bross et al., Block merging for quad tree based partitioning in HEVC. SPIE Applications of digital image processing XXXV, vol. 8499, paper 8499-, Aug 2012
E63. P. Hanhart et al., Subjective quality evaluation of the upcoming HEVC video compression standard. SPIE Applications of digital image processing XXXV, vol. 8499, paper 8499-30, Aug 2012
E64. M. Horowitz et al, Informal subjective quality comparison of video compression performance of the HEVC and H.264/MPEG-4 AVC standards for low delay applications. SPIE Applications of digital image processing XXXV, vol. 8499, paper 8499-31, Aug 2012
E65. V. Sze, M. Budagavi, High throughput CABAC entropy coding in HEVC. IEEE Trans. CSVT **22**, 1778–1791 (2012)
E66. V. Sze, A.P. Chandrakasan, Joint algorithm-architecture optimization of CABAC to increase speed and reduce area cost. IEEE ISCAS, Proceedings, pp. 1577–1580 (2011)
E67. H. Schwartz, D. Marpe, T. Wiegand, Overview of the scalable video coding extension of the H.264/AVC standard. IEEE Trans. CSVT **17**, 1103–1120 (2007) (This is a special issue on scalable video coding in H.264/AVC)
E68. JSVM—joint scalable video model reference software for scalable video coding (on line) http://ip.hhi.de/imagecom_GI/savce/downloads/SVC-Reference-software.htm
E69. H. Lakshman, H. Schwarz, T. Wiegand, Generalized interpolation based fractional sample motion compensation. IEEE Trans. CSVT **23**, 455–466 (2013). A sample C++ implementation (that can be tested using the first test model of HEVC, HM1.0) can be found in. http://phenix.int-evry.fr/jct/doc_end_user/documents/4_Daegu/wg11/JCTVC-D056-v2.zip
E70. K. Ugur et al., High performance low complexity video coding and the emerging high efficiency video coding standard. IEEE Trans. CSVT **20**, 1688–1697 (2010)
E71. A. Norkin et al., HEVC deblocking filter. IEEE Trans. CSVT **22**, 1746–1754 (2012)
E72. J. Sole et al., Transform coefficient coding in HEVC. IEEE Trans. Circuits Syst Video Technol **22**, 1765–1777 (2012)
E73. JSVM9 Joint scalable video model 9 http://ip.hhi.de/imagecom_GI/savce/downloads
E74. Z. Shi, X. Sun, F. Wu, Spatially scalable video coding for HEVC. IEEE Trans. CSVT **22**, 1813–1826 (2012)
E75. A. Krutz et al., Adaptive global motion temporal filtering for high efficiency video coding. IEEE Trans.CSVT **22**, 1802–1812 (2012)
E76. J. Lainema et al., Intra coding of the HEVC standard. IEEE Trans. CSVT **22**, 1792–1801 (2012)
E77. H. Li, B. Li, J. Xu, Rate distortion optimized reference picture management for high efficiency video coding. IEEE Trans. CSVT **22**, 1844–1857 (2012)
E78. Special issue on video coding: HEVC and beyond, IEEE Journal of selected topics in signal processing, Final manuscripts due by 25 June 2013
E79. G. Bjontegaard, Calculation of average PSNR differences between RD-Curves. ITU-T SG16, Doc. VCEG-M33, 13th VCEG meeting, Austin, TX, Apr 2001. http://wfpt3.itu.int/av-arch/video-site/0104_Aus/VCEG-M33.doc
E80. G. Bjontegaard, Improvements of the BD-PSNR model. ITU-T SG16 Q.6, Doc. VCEG-AI11, Berlin, Germany, July 2008

E81. T. Schierl et al., System layer integration of HEVC. IEEE Trans. CSVT **22**, 1871–1884 (2012)
E82. G. Correa et al., Performance and computational complexity assessment of high efficiency video encoders. IEEE Trans. CSVT **22**, 1899–1909 (2012)
E83. M. Zhou et al., HEVC lossless coding and improvements. IEEE Trans. CSVT **22**, 1839–1843 (2012)
E84. C. Yeo, Y. H. Tan, Z. Li, Dynamic range analysis in HEVC residual coding and reconstruction. IEEE Trans. CSVT **23**, 2013 (To be published)
E85. C.-M. Fu et al., Sample adaptive offset in the HEVC standard. IEEE Trans. CSVT **22**, 1755–1764 (2012)
E86. C.C. Chi et al., Parallel scalability and efficiency of HEVC parallelization approaches. IEEE Trans. CSVT **22**, 1827–1838, 2012
E87. J. Vanne et al., Comparative rate-distortion complexity analysis of HEVC and AVC video codecs. IEEE Trans. CSVT **22**, 1885–1898 (2012)
E88. Y. Yuan et al., Quadtree based non-square block structure for interframe coding in high efficiency video coding. IEEE Trans. CSVT **22**, 1707–1719 (2012)
E89. R. Sjoberg et al., Overview of high-level syntax and reference picture management. IEEE Trans. CSVT **22**, 1858–1870 (2012)
E90. P. Helle et al., "Block merging for quadtree-based partitioning in HEVC", IEEE Trans. Circuits Syst Video Technol **22**, 1720–1731 (2012)
E91. I.-K. Kim et al., Block partitioning structure in HEVC standard. IEEE Trans. CSVT **22**, 1697–1706 (2012)
E92. T. Lin et al., Mixed chroma sampling-rate high efficiency video coding for full-chroma screen content. IEEE Trans. CSVT **23**, 173–185 (2013)
E93. E. Peixoto, E. Izquierdo, A new complexity scalable transcoder from H.264/AVC to the new HEVC codec. IEEE ICIP 2012, session MP.P1.13 (Poster), Orlando, FL, Sept–Oct 2012
E94. K. Anderson, R. Sjobetg, A. Norkin, BD measurements based on MOS (online). ITU-T Q6/SG16, document VCEG-AL23, Geneva, Switzerland, July 2009. Available: http://wfpt3.itu.int/av-arch/video-site/0906_LG/VCEG-AL23.zip
E95. HEVC open source software (encoder/decoder), https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/tags/HM-6.0
E96. T. Shanableh, E. Peixoto, E. Izquierdo, MPEG-2 to HEVC transcoding with content based modeling. IEEE Trans. CSVT **23**, 2013 (To be published). References [E97]–[E101] are from VCIP 2012. Pl access http://www.vcip2012.org. There are also some papers on HEVC in the Demo session
E97. G.J. Sullivan, HEVC; The next generation in video compression. Keynote speech, Visual Communications and Image Processing, VCIP 2012, San Diego, CA 27–30 Nov 2012
E98. Two oral sessions on HEVC (W03 and F01). Also panel discussion on "All about HEVC" Moderator Shipeng Li, Microsoft Research Asia, VCIP 2012, San Diego, CA Nov 2012
E99. X. Zhang, S. Liu, S. Lei, Intra mode coding in HEVC standard. Visual Communications and Image Processing, VCIP 2012, San Diego, CA, 27–30 Nov 2012
E100. L. Yan et al., Implementation of HEVC decoder on X86 processors with SIMD optimization. Visual Communications and Image Processing, VCIP 2012, San Diego, CA, 27–30 Nov 2012
E101. W. Dai, H. Xiong, Optimal intra coding of HEVC by structured set prediction mode with discriminative learning. Visual Communications and Image Processing, VCIP 2012, San Diego, CA, 27–30 Nov 2012
E102. G. Li et al., Uniform probability model for deriving intra prediction angular table in high efficiency video coding. Visual Communications and Image Processing, VCIP 2012, San Diego, CA, 27–30 Nov 2012

E103. Y. Duan, An optimized real time multi-thread HEVC decoder. Visual Communications and Image Processing, VCIP 2012, San Diego, CA, 27–30 Nov 2012
E104. S. Gangavathi, Complexity reduction of H.264 using parallel programming. M.S. Thesis, EE Dept., UTA, Arlington, TX, 2012
E105. M.T. Pourazad et al., HEVC: the new gold standard for video compression. IEEE CE Mag **1**(3), 36–46 (2012)
E106. Y. Zhang, Z. Li, B. Li, Gradient-based fast decision for intra prediction in HEVC. Visual Communications and Image Processing, VCIP 2012, San Diego, CA, 27–30 Nov 2012
E107. H. Lv et al., A comparison of fractional-pel interpolation filters in HEVC and H.264/AVC. Visual Communications and Image Processing, VCIP 2012, San Diego, CA, 27–30 Nov 2012
E108. J. Wang et al., Multiple sign bits hiding for high efficiency video coding. Visual Communications and Image Processing, VCIP 2012, San Diego, CA, 27–30 Nov 2012
E109. C.-M. Fu et al., Sample adaptive offset for HEVC, in *Proceedings of the International workshop on multimedia signal processing, (MMSP)*, Hangzhou, China, Oct 2011
E110. S. Subbarayappa, Implementation and analysis of directional discrete cosine transform in baseline profile in H.264. M.S. Thesis, EE Dept., UTA, Arlington, Texas, 2012
E111. P. Anjanappa, Performance analysis and implementation of mode dependent DCT/DST in H.264/AVC. M.S. Thesis, EE Dept., UTA, Arlington, Texas, 2012
E112. Q. Cai et al, Losssy and lossless intra coding performance evaluation: HEVC, H.264/AVS, JPEG 2000 and JPEG LS. APSIPA, Los Angeles, CA, 2012
E113. K. Chen et al., Efficient SIMD optimization of HEVC encoder over X86 processors. APSIPA, Los Angeles, CA, Dec 2012
E114. H. Lv et al., An efficient NEON-based quarter-pel interpolation method for HEVC. APSIPA, Los Angeles, CA, Dec 2012
E115. Q. Yu et al., Early termination of coding unit splitting for HEVC. APSIPA, Los Angeles, CA, Dec 2012
E116. J.M. Nightingale, Q. Wang, C. Grecos, Priority-based methods for reducing the input of packet loss on HEVC encoded video streams. SPIE/EI, 8656-15, Feb 2013
E117. Joint preliminary call for proposals on scalable video coding extensions of high efficiency video coding (HEVC), Geneva, Switzerland, May 2012
E118. S. Kamp, M. Wien, Decoder-side motion vector derivation for block-based video coding. IEEE Trans. CSVT **22**, 1732–1745 (2012)
E119. A.Saxena, F.Fernandes, Y.Reznik, Fast transforms for Intra-prediction-based image and video coding, in *Proceedings of the IEEE Data Compression Conference (DCC'13)*, Snowbird, UT, 20–23 Mar 2013 (to appear)
E120. R. Chivukula, Y. Reznik, Fast computing of discrete cosine and sine transforms of types VI and VII, in *Applications of Digital Image Processing XXXIV, Proceedings of SPIE*, ed. by A.G. Tescher, vol. 8135 (2011), pp. 813505-1–14
E121. M. Zhou et al, HEVC lossless coding and improvements. SPIE/EI, vol. 8666-10, Burlingame, CA, Feb 2013
E122. R. Garcia, H. Kalva, Subjective evaluation of HEVC in mobile devices. SPIE/EI, vol. 8667-19, Burlingame, CA, Feb 2013
E123. B. Bross et al., High efficiency video coding (HEVC) text specification draft 10 (for FDIS & Consent). JCTVC-L1003v13, Jan 2013
E124. X. Xiu et al., Improved motion prediction for scalable extensions of HEVC. SPIE/EI, vol. 8666-5, Burlingame, CA Feb 2013
E125. T. Hinz et al, An HEVC extension for spatial and quality scalable video coding. SPIE/EI, vol. 8666-6, Burlingame, CA Feb 2013
E126. L.J. Kerofsky et al., Scalable extensions of HEVC for next generation services. SPIE/EI, vol. 8666-8, Burlingame, CA, Feb 2013

E127. ISO/IEC JTC-1/SC29/WG11 w12957, Joint call for proposals on scalable video coding extensions of high efficiency video coding, July 2012
E128. S.G. Deshpande et al., An improved hypothetical reference decoder for HEVC. SPIE/EI, vol. 8666-9, Burlingame, CA Feb 2013
E129. J. Han et al., Towards jointly optimal spatial prediction and adaptive transform in video/ image coding. IEEE ICASSP, pp. 726–729, Mar 2010
E130. A. Saxena, F. Fernanades, CE7: Mode-dependent DCT/DST for intra prediction in video coding. ITU-T / ISO-IEC Document: JCTVC-D033, Jan 2011
E131. A. Saxena, F. Fernandes, Mode dependent DCT/DST for intra prediction in block-based image/video coding. IEEE ICIP, pp. 1685–1688, Sept 2011
E132. Z. Gu et al., Fast depth modeling mode selection for 3D HEVC depth intra coding, in *IEEE ICME* 2013.
E133. M.-O. Kim, H.-J. Lee, N. Ling, Fast merge mode decision for diamond search in high efficiency video coding, in *VCIP* 2013

General

G1. T. Ebrahimi, M. Kunt, Visual data compression for multimedia applications. Proc. IEEE **86**, 1109–1125 (1998)
G2. T. Wiegand, B. Girod, Lagrange multiplier selection in hybrid video coder control. Proc. IEEE ICIP **3**, 542–545 (2001)
G3. A.M. Patino et al, 2D-DCT on FPGA by polynomial transformation in two-dimensions, in *IEEE, Proceedings of the 2004 International Symposium on Circuits and Systems, ISCAS'04*, vol. 3, pp. 365–368, May 2004
G4. G.A. Davidson et al., ATSC video and audio coding. Proc. IEEE **94**, 60–76 (2006)
G5. J. Golston, A. Rao, Video compression: system trade-offs with H.264, VC-1 and other advanced CODECs. Texas Instruments, White Paper, Nov 2006. This paper was written for and presented at the Embedded Systems Conference - Silicon Valley, San Jose, CA, Apr 2006
G6. C.-H. Yu, S.-Y. Chen, Universal colour quantisation for different colour spaces. IEE Proc. Vis. Image Sig. Proces. **153**, 445–455 (Aug. 2006)
G7. R.A. Burger et al, A survey of digital TV standards China, in *IEEE Second International Conference on Communication and Networking in China*, pp. 687–696, Aug 2007
G8. T. Wiegand, G.J. Sullivan, The picture phone is here. Really. IEEE Spectrum **48**, 50–54 (2011)
G9. N. Jayant, Frontiers of audiovisual communications: new convergences of broadband communications, computing, and rich media, in *Proceedings of the IEEE, Special Issue, Scanning the Issue*, vol. 100, pp. 821–823, Apr 2012 (Several papers on audio visual communications)
G10. L. Chiariglione, Multimedia standards: interfaces to innovation. Proc. IEEE **100**, 893–904 (2012)
G11. Digital image processing assignment website: http://www-ee.uta.edu/dip/Courses/EE5356/ ee_5356.htm.
G12. W. Bruch, The PAL colour TV transmission system. IEEE Trans. Broadcast Telev. Receivers **12**(2), 87–96 (1966)
G13. T.B. Wu, K.R. Rao, Digital TV receiver for NTSC color TV signals with dual word-length DPCM coding. IEEE Trans. Broadcast. **28**(1), 20–24 (1982)
G14. J.F. Blinn, NTSC: nice technology, super color. IEEE Comput. Graph. Appl. **13**(2), 17–23 (1993)

G15. F. Nagumo, T. Asaida, S. Ochi, CCD digital color camera. IEEE Trans. Consum. Electron. **CE-26**(3), 339–352 (1980)
G16. Graphics Display Resolution: http://en.wikipedia.org/wiki/Graphics_display_resolution
G17. B. Schweizer, The future of super high-definition televisions [video], http://www.tested.com/tech/tvs/1086-the-future-of-super-high-definition-televisions-video/

H.264 Advance Video Coding (AVC)

H1. R. Schäfer, T. Wiegand, H. Schwarz, The emerging H.264/AVC standard. EBU Tech. Rev. 1–12 (2003)
H2. JVT Draft ITU-T recommendation and final draft international standard of joint video specification (ITU-T Rec. H.264–ISO/IEC 14496-10 AVC), Mar 2003, JVT-G050, available on http://ip.hhi.de/imagecom_G1/assets/pdfs/JVT-G050.pdf
H3. T. Wiegand et al., Overview of the H.264/AVC video coding standard. IEEE Trans. CSVT, Special Issue on H.264/AVC **13**, 560–576 (2003)
H4. H.S. Malvar et al., Low-complexity transform and quantization in H.264/AVC. IEEE Trans. CSVT **13**, 598–603 (2003)
H5. T. Stockhammer, M.M. Hannuksela, T. Wiegand, H.264/AVC in wireless environments. IEEE Trans. CSVT **13**, 657–673 (2003)
H6. S. Wenger, H.264/AVC over IP. IEEE CSVT **13**, 645–656 (2003)
H7. A. Tamhankar, K.R. Rao, An overview of H.264/MPEG-4 part 10, in *Proceedings of 4th EURASIP-IEEE Conference focused on Video/Image Processing and Multimedia Communications*, vol. 1, Zegreb, Croatia, pp. 1–51, July 2003
H8. J. Ostermann et al., Video coding with H.264/AVC: tools, performance, and complexity. IEEE Circuits Syst. Mag. **4**, 7–28 (2004)
H9. M. Fieldler, Implementation of basic H.264/AVC decoder. Seminar Paper at Chemnitz University of Technology, June 2004
H10. Y. Zhang et al., Fast 4 × 4 intra-prediction mode selection for H.264, in *Proceedings of International Conference on Multimedia Expo*, Taipei, Taiwan, pp. 1151–1154, June 2004
H11. G.J. Sullivan, P. Topiwala, A. Luthra, The H.264/AVC advanced video coding standard: Overview and introduction to the fidelity range extensions, in *SPIE Conference on Applications of Digital Image Processing XXVII*, vol. 5558, pp. 454–474, Aug 2004
H12. F. Fu et al., Fast intra prediction algorithm in H.264/AVC, in *Proceedings of the 7th International Conference on Signal Process*, Beijing, China, pp. 1191–1194, Sep 2004
H13. A. Puri, X. Chen, A. Luthra, Video coding using the H.264/MPEG-4 AVC compression standard. Signal Process. Image Commun. **19**, 793–849 (2004)
H14. F. Pan et al., Fast intra mode decision algorithm for H.264/AVC video coding, in *Proc. IEEE ICIP*, pp. 781–784, Singapore, Oct 2004
H15. *Proc. IEEE*, Special issue on advances in video coding and delivery, vol. 93, pp. 3–193, Jan 2005 (This has several overview papers)
H16. T. Sikora, Trends and perspectives in image and video coding. Proc. IEEE **93**, 6–17 (2005)
H17. G.J. Sullivan, T. Wiegand, Video compression—from concepts to H.264/AVC standard. Proc. IEEE **93**, 18–31 (Jan. 2005)
H18. D. Marpe, T. Wiegand, H.264/MPEG4-AVC fidelity range extensions: tools, profiles, performance, and application areas. IEEE ICIP **1**, 593–596 (2005)
H19. Y-L. Lee, K-H. Han, Complexity of the proposed lossless intra for 4:4:4. (ISO/IEC JTC1/SC29/WG11 and ITU-T SG 16 Q.6) document JVT-Q035, 17-21 Oct 2005
H20. D. Kumar, P. Shastry, A. Basu, Overview of the H.264 / AVC, in *8th Texas Instruments Developer Conference India*, Bangalore, 30 Nov–1 Dec 2005

H21. *Proc. IEEE*, Special Issue on Global Digital Television: Technology and Emerging Services, vol. 94, Jan 2006
H22. G. Raja, M. Mirza, In-loop de-blocking filter for H.264/AVC Video, in *Proceedings of the 2nd IEEE-EURASIP Int. Symposium Communication, Control and Signal Processing 2006, ISCCSP*, Marrakech, Morocco, Mar 2006
H23. S.-K. Kwon, A. Tamhankar, K.R. Rao, Overview of H.264/MPEG-4 Part 10. J. Vis. Commun. Image Represent. (JVCIR) **17**, 186–216 (2006). Special Issue on "Emerging H.264/AVC Video Coding Standard"
H24. S. Kumar et al., Error resiliency schemes in H.264/AVC standard. J. Vis. Commun. Image Represent. (JVCIR) **17**, 425–450 (2006) (Special Issue on H.264/AVC)
H25. D. Marpe, T. Wiegand, G.J. Sullivan, The H.264/MPEG-4 AVC standard and its applications. IEEE Commun. Mag. **44**, 134–143 (2006)
H26. S. Naito, A. Koike, Efficient coding scheme for super high definition video based on extending H.264 high profile. Proc. SPIE Vis. Commun. Image Process. **6077**, 607727-1–607727-8 (2006)
H27. J. Kim et al., in *Complexity Reduction Algorithm for Intra Mode Selection in H.264/AVC Video Coding*, ed. by J. Blanc-Talon et al. ACIVS 2006, LNCS 4179 (Springer, 2006), pp. 454–465
H28. S. Ma, C.C. Kuo, High-definition video coding with supermacroblocks, in *Proceedings of SPIE VCIP*, vol. 6508, pp. 650816-1–650816-12, Jan 2007
H29. T. Wiegand, G.J. Sullivan, The H.264/AVC video coding standard. IEEE SP Mag., **24**, 148–153 (2007) (Info on H.264/AVC resources)
H30. H.264/AVC JM 18.0 reference software: http://iphome.hhi.de/suehring/tml/download/
H31. H.264 video compression standard. White paper, Axis Communications
H32. W. Lee et al., High speed intra prediction scheme for H.264/AVC. IEEE Trans. Consum. Electron. **53**(4), 1577–1582 (2007)
H33. Y.Q. Shi, H. Sun, Image and video compression for multimedia engineering, 2nd edn. (CRC Press, Boca Raton, 2008) (Chapter on H.264)
H34. B. Furht, S.A. Ahson, Handbook of mobile broadcasting, DVB-H, DMB, ISDB-T and MEDIAFLO. (CRC Press, Boca Raton, FL, 2008) (H.264 related chapters)
H35. M. Jafari, S. Kasaei, Fast intra- and inter-prediction mode decision in H.264 advanced video coding. Int. J. Comput. Sci. Netw. Secur. **8**, 130–140 (2008)
H36. A.M. Tourapis, H.264/14496-10 AVC reference software manual (2009) [Online]. Available:
http://iphome.hhi.de/suehring/tml/
JM%20Reference%20Software%20Manual%20%28JVT-AE010%29.pdf
H37. P. Carrillo, H. Kalva, T. Pin, Low complexity H.264 video encoding. SPIE, vol. 7443, paper 74430A, Aug 2009
H38. B.M.K. Aswathappa, K.R. Rao, Rate-distortion optimization using structural information in H.264 strictly intra-frame encoder, in *IEEE Southeastern Symposium on System Theory (SSST)*, Tyler, TX, Mar 2010
H39. D. Han et al., Low complexity H.264 encoder using machine learning. IEEE SPA 2010, Poznan, Poland, pp. 40–43, Sept 2010
H40. T. Sathe, Complexity reduction of H.264/AVC motion estimation using OpenMP. M.S. Thesis, EE Dept., UTA, 2011
H41. D. Han, A. Kulkarni, K.R. Rao, Fast inter-prediction mode decision algorithm for H.264 video encoder. IEEE ECTICON 2012, Cha Am, Thailand, May 2012
H42. H.264/MPEG-4 AVC: http://en.wikipedia.org/wiki/H.264
H43. S. Subbarayappa, Implementation and analysis of directional discrete cosine transform in H.264 for baseline profile. M.S. Thesis, EE Dept., UTA, May 2012

H44. I.E. Richardson, H.264 / MPEG-4 Part 10 White Paper, www.vcodex.com
H45. I.E. Richardson, Overview: What is H.264? (2011), www.vcodex.com
H46. I.E. Richardson, A Technical Introduction to H.264/AVC (2011) [Online], Available: http://www.vcodex.com/files/H.264_technical_introduction.pdf
H47. I.E. Richardson, White Paper: H.264 / AVC intra prediction (2011), www.vcodex.com
H48. I.E. Richardson, White Paper: H.264/AVC inter prediction (2011), www.vcodex.com
H49. I.E. Richardson, White Paper: 4 × 4 transform and quantization in H.264/AVC (2010), www.vcodex.com
H50. I.E. Richardson, White Paper: H.264/AVC loop filter (2011) www.vcodex.com
H51. Joint Video Team (JVT), ITU-T website, http://www.itu.int/ITU-T/studygroups/com16/jvt/
H52. RGB Spectrum's H.264 codecs hit the mark in the latest U.S. missile test program (2012), www.rgb.com
H53. ISO website: http://www.iso.org/iso/home.htm
H54. IEC website: http://www.iec.ch/
H55. ITU-T website: http://www.itu.int/ITU-T/index.html
H56. K.V.S. Swaroop, K.R. Rao, Performance Analysis and Comparison of JM 15.1 and Intel IPP H.264 Encoder and Decoder, IEEE 2010 42nd Southeastern Symposium on System Theory (SSST), Tyler, TX, pp. 371–375, Mar 2010
H57. S.K. Muniyappa, "Performance analysis and comparison of JM, Intel IPP and × 264 for H.264 Software", MS Thesis, EE Dept. UTA (2011)
H58. W. Zhao, J. Fan, A. Davari, H.264-based wireless surveilance sensors in application to target identification and tracking, i-manager's J. Softw. Eng. **4** (2) (2009). Website: http://web.eng.fiu.edu/fanj/pdf/j5_i-manager09h264_camera.pdf
H59. W. Zhao et al., H.264-based architecture of digital surveillance network in application to computer visualization, i-manager's J. Softw. Eng. **4** (4) (2010). Website: http://web.eng.fiu.edu/fanj/pdf/J8_i-mgr10architecture_camera.pdf
H60. C. Deng et al., Performance analysis, parmeter selection and extensions to H.264/AVC FRExt for high resolution video coding, J. Vis. Commun. Image R., **22** (in press), Available online, Feb 2011.

Integer DCT

I1. N. Ahmed, T. Natarajan, K.R. Rao, Discrete cosine transform. IEEE Trans. Comput. **C-23**, 90–93 (1974)
I2. W. Chen, C.H. Smith, S.C. Fralick, A fast computational algorithm for the discrete cosine transform. IEEE Trans. Commun. **COM-25**, 1004–1009 (1977)
I3. M. Vetterli, A. Ligtenberg, A discrete Fourier-cosine transform chip. IEEE J. Sel. Areas Commun. **4**, 49–61 (1986)
I4. W.K. Cham, R.J. Clarke, Application of the principle of dyadic symmetry to the generation of orthogonal transform. IEE Proc. F: Commun. Radar Signal Process. **133**(3), 264–270 (1986)
I5. C. Loeffler, A. Lightenberg, G. Moschytz, Practical fast 1-D DCT algorithms with 11 multiplications. Proc. IEEE ICASSP **2**, 988–991 (1989)
I6. W.K. Cham, Development of integer cosine transforms by the principle of dyadic symmetry. IEE Proc. I: Commun. Speech Vis. **136**(4), 276–282 (1989)
I7. W.K. Cham, Y.T. Chan, An order-16 integer cosine transform. IEEE Trans. Signal Process. **39**, 1205–1208 (1991)
I8. F. Bruekers, A. Enden, New networks for perfect inversion and perfect reconstruction. IEEE J. Sel. Areas Commun. **10**, 130–137 (1992)

I9. W. Sweldens, The lifting scheme: a custom-design construction of biorthogonal wavelets. Appl. Comput. Harmon. Anal. **3**(2), 186–200 (1996)
I10. I. Daubechies, W.A. Pearlman, Factoring wavelet transforms into lifting steps. J. Fourier Anal. Appl. **4**, 247–269 (1998)
I11. J. Liang, T. Tran, P. Topiwala, A 16-bit architecture for H.26L, treating DCT transforms and quantization. ITU VCEG doc M16, Austin, TX, Apr 2001
I12. J. Liang, T.D. Tran, Fast multiplierless approximations of the DCT with the lifting scheme. IEEE Trans. Signal Process. **49**, 3032–3044 (2001)
I13. C.X. Zhang et al., The technique of pre-scaled transform. IEEE ISCAS **1**, 316–319 (2005)
I14. J. Dong et al., A new approach to compatible adaptive block-size transforms. Visual Communication and Image Processing 2005, in *Proceedings of SPIE*, vol. 5960, pp., July 2005
I15. G.J. Sullivan, Standardization of IDCT approximation behavior for video compression: the history and the new MPEG-C parts 1 and 2 standards, in *Proceedings of SPIE*, vol. 6696, pp. 669611 1–22, 2007
I16. J. Dong et al., A universal approach to developing fast algorithm for simplified order-16 ICT. IEEE ISCAS, pp. 281–284, June 2007
I17. C.X. Zhang et al., The technique of prescaled integer transform: concept, design and applications. IEEE Trans. CSVT **18**, 84–97 (2008)
I18. C.-P. Fan, G.-A. Su, Efficient fast 1-D 8×8 inverse integer transform for VC-1 application. IEEE Trans. CSVT **19**, 1–7 (2009)
I19. J. Dong et al., 2D order-16 integer transforms for HD video coding. IEEE Trans. CSVT **19**, 1462–1474 (2009)
I20. W.K. Cham, C. Fong, Simple order-16 integer transform for video coding. IEEE ICIP 2010, Hong Kong, pp. 161–164, Sept 2010
I21. M.P. Krishnan, Implementation and performance analysis of 2-D order 16 integer transforms in H.264/AVC and AVS-video for high definition video coding. M.S. Thesis, EE Dept., UTA, TX, Dec 2010

JPEG 2000

J1. M.D. Adams, The JPEG-2000 still image compression standard. JPEG Tutorial download from website: http://www.ece.uvic.ca/~mdadams/jasper/ (also software). JPEG-2000 open source software (Jasper Version 1.900.0)
J2. Z. Xiong et al., A comparative study of DCT- and wavelet-based image coding. IEEE Trans. Circuits Syst. Video Tech. **9**, 692–695 (1999)
J3. T. Fukuhara et al., Motion-JPEG2000 standardization and target market. IEEE ICIP **2**, 57–60 (2000)
J4. D. Santa-Cruz et al., JPEG 2000 still image coding versus other standards, in *Proceedings of the SPIE's 45th annual meeting, Applications of Digital Image Processing XXIII*, vol. 4115, 2000
J5. M.D. Adams, F. Kossentini, Jasper: a software-based JPEG-2000 codec implementation, in *Proceedings of IEEE International Conference Image Processing,* vol. 2, Vancouver, BC, Canada, pp. 53–56, Oct 2000
J6. C. Christopoulous, A. Skodras, T. Ebrahimi, The JPEG 2000 still image coding system: an overview. IEEE Trans. Consum. Electron. **46**, 1103–1127 (2000)
J7. M.S. Zhong, Z.M. Ma, JPEG 2000 based scalable reconstruction of image local regions. IEEE ISIMP 2001, Hong Kong, May 2001
J8. D.S. Taubman, M.W. Marcellin, JPEG 2000: image compression fundamentals, standards and practice, Kluwer, 2001
J9. A. Skodras, C. Christopoulous, T. Ebrahimi, JPEG-2000: the upcoming still image compression standard. Pattern Recogn. Lett. **22**, 1337–1345 (2001)

J10. A. Skodras, C. Christopoulus, T. Ebrahimi, The JPEG2000 still image compression standard. IEEE Signal Process. Mag. **18**, 36–58 (2001)
J11. Special Issue on JPEG-2000, Signal Processing: Image Communication, vol. 17 (2002), pp. 1–144
J12. M. Rabbani, R. Joshi, An overview of the JPEG 2000 still image compression standard. Signal Process. Image Commun. **17**, 3–48 (2002)
J13. J. Hunter, M. Wylie, JPEG2000 Image compression: a real time processing challenge. Adv Imaging **18**, 14–17 and 43 (2003)
J14. A. Stoica, C. Vertan, C. Fernandez-Maloigne, Objective and subjective color image quality evaluation for JPEG 2000-compressed images. IEEE Int. Symp. Signals Circuits Syst. **1**, 137–140 (2003)
J15. Special section on JPEG 2000 digital imaging. IEEE Trans. Consum. Electron. **49** (2003)
J16. D. Marpe et al., Performance evaluation of motion JPEG2000 in comparison with H.264/ operated in intra-coding mode. Proc. SPIE **5266**, 129–137 (2004)
J17. D. Marpe, V. George, T. Wiegand, Performance comparison of intra-only H.264/AVC HP and JPEG 2000 for a set of monochrome ISO/IEC test images. JVT-M014, pp.18–22, Oct 2004
J18. D.T. Lee, JPEG 2000: retrospective and new developments. Proc. IEEE **93**, 32–41 (Jan. 2005)
J19. P. Topiwala, Comparative study of JPEG2000 and H.264/AVC FRExt I-frame coding on high definition video sequences, in *Proceedings of SPIE Internatioal Symposium, Digital Image Processing*, San Diego, CA, Aug 2005
J20. M. Ouaret, F. Dufaux, T. Ebrahimi, On comparing JPEG 2000 and intraframe AVC, in *SPIE, Applications of Digital Image Processing XXIX*, vol. 6312, Aug 2006
J21. P. Topiwala, T. Tran, W. Dai, Performance comparison of JPEG2000 and H.264/AVC high profile intra-frame coding on HD video sequences, in *Proceedings of SPIE Internatioal Symposium, Applications of Digital Image Processing XXIX*, vol. 6312, San Diego, CA, Aug 2006. JPEG XR (HD photo of Microsoft)
J22. T. Tran, L. Liu, P. Topiwala, Performance comparison of leading image codecs: H.264/ AVC intra, JPEG 2000, and Microsoft HD photo, in *Proceedings of SPIE International Symposium*, *Applications of Digital Image Processing XXX*, vol. 6696, San Diego, CA, Sept 2007
J23. M.D. Adams, JasPer software reference manual (Version 1.900.0). ISO/IEC JTC 1/SC 29/ WG 1 N 2415, Dec 2007
J24. S-T. Hsiang, A new subband/wavelet framework for AVC/H.264 intraframe coding and performance comparison with motion-JPEG 2000, in *VCIP, Proceedings of SPIE-IS& T Electronic Imaging, SPIE,* vol. 6822, pp. 68220P-1–68220P-12, Jan 2008
J25. P. Schelkens, A. Skodras, T. Ebrahimi, *The JPEG 2000 suite* (Wiley, Hoboken, 2009)
J26. J.A. Dickerson http://www.eng.iastate.edu/ee528/SpecialTopics/JPEG2000/jpeg2000.htm. DIP course website: http://www.eng.iastate.edu/ee528/
J27. C. Bako, JPEG 2000 image compression (2006), http://broadcastengineering.com/mag/ broadcasting_jpeg_image_compression/
J28. JPEG 2000 links (Include documents) http://www.jpeg.org/jpeg2000/

JPEG-LS (Lossless), LOCO-I (Low Complexity Lossless Compression for Images)

JL1. JPEG-LS reference software website http://www.hpl.hp.com/loco/
JL2. M.J. Weinberger, G. Seroussi, G. Sapiro, *LOCO-I: A Low Complexity, Context-Based, Lossless Image Compression Algorithm*. Hewlett-Packard Laboratories, Palo Alto
JL3. M.J. Weinberger, G. Seroussi, G. Sapiro, LOCO-I A low complexity context-based, lossless image compression algorithm. IEEE Proceedings of 1996 DCC, Snowbird, Utah, pp. 140–149, Mar 1996

JL4. M.J. Weinberger, G. Seroussi, G. Sapiro, The LOCO-I lossless image compression algorithm: principles and standardization into JPEG-LS. IEEE Trans. Image Process. **9**, 1309–1324 (2000)
JL5. JPEG-LS Compression Encoder Core, Alma Tech., http://www.cast-inc.com/ip-cores/images/jpegls-e/index.html (ISO-14495-1/ITU-T.87)

JPEG

JP1. G.K. Wallace, The JPEG still picture compression standard. Commun. ACM **34**, 31–44 (1991)
JP2. Joint Photographic Experts Group, JPEG: http://www.jpeg.org/
JP3. JPEG, http://en.wikipedia.org/wiki/JPEG
JP4. MJPEG, http://en.wikipedia.org/wiki/MJPEG
JP5. E.L. Tan, W.S. Gan, Perceptually tuned subband coder for JPEG. J. Real Time Image Process. **6**, 101–115 (2011)
JP6. W.B. Pennebaker, J.L. Mitchell, The JPEG data compression standard for continuous tone still images. Van Nostrand Reinhold (1992)
JP7. JPEG reference website, http://www-ee.uta.edu/Dip/Courses/EE5359/JPEG_1.pdf
JP8. JPEG reference website, http://www-ee.uta.edu/Dip/Courses/EE5359/JPEG_2.pdf
JP9. JPEG coding http://ijg.org/
JP10. JPEG reference software website, ftp://ftp.simtel.net/pub/simtelnet/msdos/graphics/jpegsr6.zip

JPEG-XR (HD Photo of Microsoft)

JX1. T. Tran, L. Liu, P. Topiwala, Performance comparison of leading image codecs: H.264/AVC intra, JPEG 2000, and Microsoft HD photo, in *Proceedings of SPIE International Symposium, Applications of Digital Image Processing XXX*, vol. 6696, San Diego, Sept 2007
JX2. S. Srinivasan et al., HD Photo: a new image coding technology for digital photography, in *Proceedings of SPIE*, vol. 6696, 66960A-1 to 66960A-19, 2007
JX3. G.J. Sullivan, ISO/IEC 29199-2 (JpegDI part 2 JPEG XR image coding—Specification). ISO/IEC JTC 1/SC 29/WG1 N 4492, Dec 2007
JX4. Microsoft HD Photo Specification, http://www.microsoft.com/whdc/xps/wmphotoeula.mspx
JX5. V.Testoni et al., On the adaptive coefficient scanning of JPEG XR/HD photo. *IEEE Data Compression Conference (DCC)*, pp. 69–78, Mar 2010
JX6. Y. Gao, D. Chan, J. Liang, JPEG-XR optimization with graph-based SOFT quantization. *IEEE ICIP 2011*, Brussels, Aug 2011

Audio Quality

L1. P. Mermelstein, Evaluation of a segmental SNR measure as an indicator of the quality of ADPCM coded speech. J. Acoust. Soc. Am. **66**(6), 1664–1667 (1979)
L2. J.D. Gibson, W.-W. Chang, Objective and subjective optimization of APC system performance. IEEE Trans. Acoust. Speech Signal Process. **38**, 1053–1058 (1990)
L3. H. Fuchs, W. Hoeg, D. Meares, ISO/MPEG subjective tests on multichannel audio systems: design and methodology, in *International Broadcasting Convention* (*IBC* 1994), pp. 152–157, 16–20 Sep 1994

L4. ITU-R BS.1116, *Methods for the Subjective Assessment of Small Impairments in Audio Systems Including Multichannel Sound System*s (1997)
L5. ITU-R BS.1284, *Method for Subjective Assessment of Sound quality—General Requirements* (1997)
L6. T. Grusec, L. Thibault, G. Soulodre, EIA/NRSC DAR systems subjective tests. I: audio codec quality. IEEE Trans. Broadcast. **43**(3), 261–267, Sept 1997
L7. ITU-R BS.1387, Method for objective measurement of perceived audio quality (1998)
L8. ITU-R BS.1534, Method for the subjective assessment of intermediate quality level of coding systems (2001)
L9. C.D. Creusere, Understanding perceptual distortion in MPEG scalable audio coding. IEEE Trans. Speech Audio Process. **13**(3), 422–431 (2005)
L10. M. Salovarda, I. Bolkovac, H. Domitrovic, Estimating perceptual audio system quality using PEAQ algorithm, in *18th Internatioal Conference on Applied Electromagnetics and Communications* (*ICECom* 2005), pp. 1–4, 12–14 Oct 2005
L11. R. Huber, B. Kollmeier, PEMO-Q – A new method for objective audio quality assessment using a model of auditory perception. IEEE Trans. Audio Speech Lang. Process. **14**(6), 1902–1911 (2006)
L12. D. Satyanarayana, K.S. Prasad, T.J.C Prasad, Quality assessment of efficient LP analysis methods, in *International Conference on Computational Intelligence and Multimedia Applications 2007*, vol. 4, pp. 423–427, 13–15 Dec 2007
L13. C.D. Creusere, K.D. Kallakuri, R. Vanam, An objective metric of human subjective audio quality optimized for a wide range of audio fidelities. IEEE Trans. Audio Speech Lang. Process. **16**, 129–136 (2008)
L14. S. Kandadai, J. Hardin, C.D. Creusere, Audio quality assessment using the mean structural similarity measure, in *IEEE ICASSP,* pp. 221–224, Mar 31–Apr 4 2008
L15. D. Palomar et al., Objective assessment of audio quality. *Signals and Systems Conference* (*ISSC* 2008), IET Irish, pp. 37–42, 18–19 June 2008
L16. C. Zhang, R.M. Hu, Subjective evaluation of sound quality for mobile spatial digital audio, in *International Conference on Computer Science and Software Engineering,* vol. 3, pp. 250–253, 12–14 Dec 2008
L17. J.C. Hardin, C.D. Creusere, A temporally varying objective audio quality metric, in *IEEE 13th Digital Signal Processing Workshop and 5th IEEE Signal Processing Education Workshop* (*DSP/SPE* 2009), pp. 21–25, 4–7 Jan 2009
L18. A.E. Mahdi, D. Picovici, Advances in voice quality measurement in modern, telecommunications. Digital Signal Process. **19**(1), 79–103 (Jan. 2009)
L19. D. Campbell, E. Jones, M. Glavin, Audio quality assessment techniques—A review, and recent developments. Signal Process. **89**, 1489–1500 (Aug. 2009)

Diretional (DDCT)

M1. H. Xu, J. Xu, F. Xu, Lifting-based directional DCT–like transform for image coding, IEEE Trans. Circuits Syst. Video Technol. **17**(10), 1325–1335, (2007)
M2. B. Zeng, J. Fu, Directional discrete cosine transforms—A new framework for image coding, IEEE Trans. Circuits Syst. Video Technol. **18**(3), 305–313 (2008)
M3. C.L. Chang, B. Girod, Direction-adapitive partitioned block transform for image coding, 15th IEEE International Conference on Image Processing, pp. 145–148, Oct 2008
M4. F. Kamisli, J.S. Lim, Transforms for motion compensation residual, IEEE ICASSP, pp. 789–792, Apr 2009
M5. F. Kamisli and J.S. Lim, Video compression with 1-d directional transforms in H.264/AVC, IEEE ICASSP, pp. 738–741, Mar 2010

M6. Intra coding with directional DCT and directional DWT, Document: JCTVC-B107_r1 Website: http://wftp3.itu.int/av-arch/jctvc-site/2010_07_B_Geneva/JCTVC-B107.zip
M7. B. Chen, H. Wang, L. Cheng, Fast directional discrete cosine transform for image compression, Opt. Eng. **49**(2), 020101, Feb 2010
M8. J. Xu, B. Zeng, F. Wu, An overview of directional transforms in image coding *Proceedings of 2010 IEEE International Symposium on Circuits and Systems*, pp. 3036–3039, Aug 2010

AVS China: Audio

N1. R.M. Hu, Y. Zhang, H.J. Ai, Digital audio compression technology and AVS audio standard research, in *IEEE Proceedings of International Symposium on Intelligent Signal Processing and Communication Systems, ISPACS 2005*, pp. 757–759, Hong Kong, China, Dec 2005
N2. R.M. Hu et al., AVS generic audio coding, in *IEEE Sixth Internationsal Conference on Parallel and Distributed Computing, Applications and Technologies, PDCAT 2005*, pp. 679–683, Dalian, China, Dec 2005
N3. L. Miao et al., Context-dependent bitplane coding in China AVS audio, in *IEEE Proceedings of 2005 International Symposium on Intelligent Signal Processing and Communication Systems, ISPACS 2005*, Hong Kong, China, pp. 765–768, Dec 2005
N4. H.J. Ai, S.X. Chen, R.M. Hu, Introduction to AVS Audio. J. Comput. Sci. Technol. **21**, 360–365 (2006)
N5. C. Zhang, R.M. Hu, H.J. Ai, AVS digital audio processing technology, in *IEEE First International Conference on Innovative Computing, Information and Control, ICICIC'06*, vol. 2, Beijing, China, pp. 342–345, Aug 2006
N6. C. Shuixian et al., Optimization of an AVS audio decoder on DSP, in *IEEE International Conference on Wireless Communication, Networking and Mobile Computing, WiCOM 2006*, Wuhan, China, pp. 1–4, Sept 2006
N7. J. Lan, T.-J. Huang, J.-H. Qu, A perception-based scalable encryption model for AVS audio, in *IEEE ICME*, Beijing, China, pp. 1778–1781, July 2007
N8. S.X. Chen et al., A window switching algorithm for AVS audio coding, in *IEEE International Conference on Wireless Communication Networking and Mobile Computing, WiCom 2007*, Shanghai, China, pp. 2889–2892, Sept 2007
N9. B.Y. Lei, K.-T. Lo, H. Lei, Optimization of AVS audio decoder implemented on RISC Core, in *IEEE ICCEE, International Conference on Computer and Electrical Engineering*, Hong Kong, China, pp. 32–35, Dec 2008
N10. China National Standard: Information technology Advanced coding of audio and video Part 3: Audio

Audio Coders

O1. H.S. Black, J.O. Edson, Pulse code modulation. Trans. Am. Inst. Electr. Eng. **66**, 895–899 (1947)
O2. K. Brandenburg, M. Bosi, Overview of MPEG audio: current and future standards for low bit-rate audio coding. JAES **45**, 4–21 (1997)
O3. R.J. van de Plassche, H.J. Schouwenaars, A monolithic high-speed sample-and-hold amplifier for digital audio. IEEE J. Solid State Circuits **18**(6), 716–722 (Dec. 1983)
O4. Y. Iso et al., 16-bit A/D converter and D/A converter for digital audio. IEEE Trans. Consum. Electron. **32**(4), 734–742 (1986)
O5. J.P. Princen, A.B. Bradley, Analysis/synthesis filter bank design based on time domain aliasing cancellation. IEEE Trans. ASSP **34**, 1153–1161 (Oct. 1986)

O6. J.P. Princen, A.W. Johnson, A.B. Bradley, Subband/transform coding using filter bank designs based on time domain aliasing cancellation. *IEEE ICASSP*, Dallas, TX, pp. 2161–2164, Apr 1987
O7. Y. Tani et al., 20 bit A-to-D & D-to-A converting systems for digital audio. IEEE Trans. Consum. Electron. **35**(3), 528–535 (1989)
O8. P. Mermelstein, G.722, a new CCITT coding standard for digital transmission of wideband audio signals. IEEE Commun. Mag. **26**, 8–15 (1988)
O9. ISO/IEC JTCl/SC29/WG11 MPEG, IS11172-3, Information technology—coding of moving pictures and associated audio for digital storage media at up to about 1.5 Mbit/s, part 3: audio. (MPEG-1) (1992)
O10. ISO/IEC JTCl/SC29/WG11 MPEG, IS13818-3 Information technology—generic coding of moving pictures and associated audio, part 3: audio. (MPEG-2) (1994)
O11. S. Shlien, Guide to MPEG-1 audio standard. IEEE Trans. Broadcast. **40**(4), 206–218 (1994)
O12. D. Pan, A tutorial on MPEG/audio compression. IEEE Multimedia **2**(2), 60–74 (1995)
O13. M. Paraskevas, J. Mourjopoulos, A differential perceptual audio coding method with reduced bitrate requirements. IEEE Trans. Speech Audio Process. **3**(6), 490–503 (1995)
O14. ITU-T Recommendation G.723.1: dual rate speech coder for multimedia communications transmitting at 5.2 and 6.3 kbits/s (1996)
O15. ITU-T Recommendation G.729: coding of speech at 8 kbit/s using conjugate-structure algebraic-code-excited linear-prediction (CS-ACELP) (1996)
O16. MPEG–2 advanced audio coding, AAC, International Standard IS 13818–7, ISO/IEC JTC1/SC29 WG11 (1997)
O17. S. Li et al., An AC-3/MPEG multi-standard audio decoder IC, in *Proceedings of the IEEE 1997 Custom Integrated Circuits Conference*, pp. 245–248, 5–8 May 1997
O18. T. Painter, A. Spanias, A review of algorithms for perceptual coding of digital audio signals, in *Proceedings IEEE 13th International Conference on Digital Signal Processing*, vol. 1, pp. 179–208, 2–4 July 1997
O19. P. Noll, MPEG digital audio coding. IEEE Signal Process. Mag. **14**(5), 59–81 (1997)
O20. Y. Shen, H.M. Ai, C.-C.J. Kuo, A progressive algorithm for perceptual coding of digital audio signals, in *Conference Record of the Thirty-Third Asilomar Conference on Signals, Systems, and Computers*, vol. 2, pp. 1105–1109, 24–27 Oct 1999
O21. T. Painter, A. Spanias, Perceptual coding of digital audio. Proc. IEEE **88**, 451–515 (2000)
O22. M. Dietz, S. Meltzer, CT-aacPlus—A state of the art audio coding scheme. EBU Tech. Rev. 1–7 (2002)
O23. P. Ekstrand, Bandwidth extension of audio signals by spectral band replication. 1st IEEE Benelux Workshop on Model based Processing and Coding of Audio (MPCA-2002), Leuven, Belgium, pp. 53–58, Nov 2002
O24. M. Bosi, M. Goldberg, *Introduction to Digital Audio Coding and Standards* (Kluwer, Boston, 2003)
O25. C.D. Creusere, Understanding perceptual distortion in MPEG scalable audio coding. IEEE Trans. Speech Audio Process. **13**(3), 422–431 (2005)
O26. S. Torres-Guijarro et al., Inter-channel de-correlation for perceptual audio coding. Appl. Acoust. **66**(8), 889–901 (2005)
O27. X.P. Hu, G.M. He, X.P. Zhou, PEAQ-based psychoacoustic model for perceptual audio coder, in *The 8th International Conference Advanced Communication Technology* (*ICACT* 2006), vol. 3, pp. 1820–1823, 20–22 Feb 2006
O28. H. Najaf-Zadeh, P. Kabal, Perceptual coding of narrow-band audio signals at low rates. IEEE Trans. Audio Speech Lang. Process. **14**(2), 609–622 (2006)
O29. M.T. Ali, M.S. Mian, Phase randomization-a new concept in perceptual audio coding, in *2006 IEEE Region 10 Conference* (*TENCON* 2006), pp. 1–4, 14–17 Nov 2006

O30. J.D. Johnston, Perceptual audio coding—A history and timeline, in *Conference Record of the Forty-First Asilomar Confernece on Signals, Systems and Computers* (*ACSSC* 2007), pp. 2085–2087, 4–7 Nov 2007
O31. T. Ogunfunmi, M. Narasimha, *Principles of Speech Coding* (CRC Press, Boca Raton, 2010)
O32. DVD-Audio Tutorial—High resolution stereo and multi-channel music, online available: http://www.timefordvd.com/tutorial/DVD-AudioTutorial.shtml
O33. DVD-Audio, online available: http://en.wikipedia.org/wiki/DVD-Audio
O34. MPEG Layer II: http://en.wikipedia.org/wiki/MPEG-1_Audio_Layer_II
O35. MP3 / MPEG Layer III: http://en.wikipedia.org/wiki/MP3
O36. FLAC—Free Lossless Audio Codec: http://flac.sourceforge.net/
O37. Vorbis: http://www.vorbis.com/
O38. A/52 B ATSC, Digital Audio Compression Standard, http://www.atsc.org/cms/standards/a_52-2010.pdf
O39. Reference software for AAC encoder—FAAC and AAC decoder—FAAD. www.audiocoding.com

VP 6

P1. Adobe press release, Macromedia and Sorenson bring video to Macromedia Flash content and applications (2002), http://www.adobe.com/macromedia/proom/pr/2002/flash_mx_video.html
P2. T. Uro, The quest for a new video codec in Flash 8, 13 Aug 2005
P3. On2 Technologies, Inc., White paper–On2 VP6 for Flash 8 Video, 12 Sept 2005
P4. On2 Technologies, Inc., White Paper—TrueMotion VP7 Video Codec (2005)
P5. VP6 bitstream and decoder specification. On2 Technologies Inc., Aug 2006
P6. C. Holder, H. Kalva, H.263 to VP6 video transcoder. *SPIE-IS&T VCIP,* vol. 6822, San Jose, CA, pp. 68222B-1–68222B-4, Jan 2008
P7. P. Wilkins, The On2 VP6 codec: how it works. EE Times, Oct 2008
P8. J. Padia, Complexity reduction for VP6 to H.264 transcoder using motion vector reuse. M.S. Thesis, 2010 EE Dept., UTA, Arlington, TX
P9. E. Ohwovoriole, Y. Andreopoulos, Rate-distortion performance of contemporary video codecs: Comparison of Google/WebM VP8, AVC/H.264, and HEVC TMuC, in *London Communication Symposium (LCS)*, London, England, Sept 2010. (VP8) http://www.ee.ucl.ac.uk/lcs/
P10. F. De Simone et al., Performance analysis of VP8 image and video compression based on subjective evaluations. SPIE Appl. Digital Image Process. XXXIV, Aug 2011 (VP8)
P11. Website of ON 2 Technologies: http://www.on2.com/
P12. J. Bankoski et al., Towards a next generation open-source video codec. SPIE/EI, vol. 8666-7, Burlingame, CA, Feb 2013
P13. J. Bankoski, P. Wilkins, Y. Xu, Technical overview of VP8, an open source video codec for the web. IEEE ICME, pp. 1–6, July 2011
P14. D. Mukherjee et al., The latest open-source video codec VP9 - An overview and preliminary vesults, PCS 2013, San Jose, CA, Dec 2013

Performance Comparison

PC1. A. Hallapuro, V. Lappalainen, T.D. Hamalainen, Performance analysis of low bit rate H.26L video encoder. IEEE ICASSP **2**, 1129–1132 (2001)

PC2. X. Wang, D. Zhao, Performance comparison of AVS and H.264/AVC video coding standards. J. Comput. Sci. Technol. **21**, 310–314 (2006). (Springer, Special Section on China AVS Standard)

PC3. B. Tang et al., AVS encoder performance and complexity analysis based on mobile video communication, in *WRI Internatioal Conference on Communication and Mobile Computing, CMC'09*, vol. 3, pp. 102–107, 6–8 Jan 2009

PC4. J. Xia et al., Perceivable artifacts in compressed video and their relation to video quality. Signal Process. Image Commun. **24**, 548–556 (2009)

Video Image Quality

Q1. M.P. Eckert, A.P. Bradley, Perceptual quality metrics applied to still image compression. Signal Process. **70**, 177–200 (1988)

Q2. X. Ran, N. Farvardin, A perceptually-motivated three-component image model, part I: description of the model. IEEE Trans. Image Process. **4**(4), 401–415 (1995)

Q3. A.M. Eskicioglu, P.S. Fisher, Image quality and measures and their performance. IEEE Trans. Commun. **43**, 2959–2965 (1995)

Q4. S. Olsson, M. Stroppiana, J. Bafna, Objective methods for assessment of video quality: state of the art. IEEE Trans. Broadcast. **43**(4), 487–495 (1997)

Q5. Special issue on image and video quality metrics. Signal Process. **70**, 153–294 (1998)

Q6. J.B. Maretens, L. Meesters, Image dissimilarity. Signal Process. **70**, 155–176 (1998)

Q7. V.R. Algazi, N. Avadhanam, R.R. Estes, Quality measurement and use of preprocessing in image compression. Signal Process. **70**, 215–229 (1998)

Q8. Z. Wang, A.C. Bovik, A universal image quality index. IEEE Signal Process. Lett. **9**, 81–84 (2002)

Q9. Z. Wang, A.C. Bovik, L. Liu, Why is image quality assessment so difficult? IEEE ICASSP **4**, 3313–3316 (2002)

Q10. Z. Wang, E.P. Simoncelli, A.C. Bovik, Multiscale structural similarity for image quality assessment, in *Conference Record of the Thirty-Seventh IEEE Asilomar Conference on Signals, Systems and Computers 2003*, vol. 2, Pacific Grove, CA, pp. 1398–1402, 9–12 Nov 2003

Q11. J.L. Li, G. Chen, Z.R. Chi, Image coding quality assessment using fuzzy integrals with a three-component image model. IEEE Trans. Fuzzy Syst. **12**, 99–106 (2004)

Q12. Z. Wang, L. Lu, A. C. Bovik, Video quality assessment based on structural distortion measurement. Signal Process. Image Commun. **19**, 121–132 (2004) (Special Issue on "Objective video quality metrics")

Q13. Z. Wang et al., Image quality assessment: From error visibility to structural similarity. IEEE Trans. Image Process. **13**, 600–612 (2004). Please access the web site http://www.ece.uwaterloo.ca/~z70wang/research/ssim/ to download MATLAB code, ssim_index.m

Q14. M.H. Pinson, S. Wolf, A new standardized method for objectively measuring video quality. IEEE Trans. Broadcast. **50**(3), 312–322 (2004)

Q15. T.N. Pappas, R.J. Safranek, Perceptual criteria for image quality evaluation, in *Handbook of Image and Video Processing*, ed. by A.C. Bovik, 2nd edn. (Academic press, 2005)

Q16. X. Shang, Structural similarity based image quality assessment: pooling strategies and applications to image compression and digit recognition. M.S. Thesis, EE Dept., UTA, Arlington, TX, 2006

Q17. R. Feghali et al., Video quality metric for bit rate control via joint adjustment of quantization and frame rate. IEEE Trans. Broadcast. **53**(1), 441–446 (2007)

Q18. Q. Huynh-Thu, M. Ghanbari, Scope of validity of PSNR in image/video quality assessment. IET Electron. Lett. **44**(13), 800–801 (2008)
Q19. Z. Li, A.M. Tourapis, New video quality metrics in the H.264 reference software, input Document to JVT, Hannover, DE, 20–25 July 2008
Q20. W.S. Kim, Z. Li, P. Pahalawatta, A.M. Tourapis from Dolby Lab, Burbank, CA, JVT-AB31, JVT Hannover (Germany) meeting, April 2008
Q21. S. Winkler, P. Mohandas, The evolution of video quality measurement: from PSNR to hybrid metrics. IEEE Trans. Broadcast. **54**(3), 660–668 (2008)
Q22. Z. Wang, A.C. Bovik, Mean square error: love it or leave it?—A new look at signal fidelity measures. IEEE Signal Process. Mag. **26**, 98–117 (2009)
Q23. A. Ninassi et al., Considering temporal variations of spatial visual distortions in video quality assessment. IEEE J. Sel. Top. Signal Process. **3**(2), 253–265 (2009)
Q24. T. Liu et al., A novel video quality metric for low bit-rate video considering both coding and packet-loss artifacts. IEEE J. Sel. Top. Signal Process. **3**(2), 280–293 (2009)
Q25. T. Oelbaum, C. Keimel, K. Diepold, Rule-based no-reference video quality evaluation using additionally coded videos. IEEE J. Sel. Top. Signal Process. **3**(2), 294–303 (2009)
Q26. M. Naccari, M. Tagliasacchi, S. Tubaro, No-reference video quality monitoring for H.264/AVC coded video. IEEE Trans. Multimedia **11**(5), 932–946 (2009)
Q27. C. Li, A.C. Bovik, Content-weighted video quality assessment using a three-component image model. J. Electron. Imaging **19**, 65–71 (2010)
Q28. L. Zhang, L. Zhang, X. Mou, D. Zhang, FSIM: a feature similarity index for image quality assessment. IEEE Trans. Image Process. **20**, 2378–2386 (2011)
Q29. C. Chukka, A universal image quality index and SSIM comparison. Report for EE 5359: Multimedia Processing, EE Dept., UTA, Arlington, TX, 2012 [Online]. Available: http://www-ee.uta.edu/Dip/Courses/EE5359/chaitanyaee5359d.pdf
Q30. MSU video quality measurement tool: http://compression.ru/video/quality_measure/video_measurement_tool_en.html
Q31. J.S. Park, T. Ogunfunmi, *A new approach for image quality assessment* (IEEE ICIEA, Singapore, 2012)
Q32. MATLAB source code for BD PSNR and BD Rate: http://www.mathworks.com/matlabcentral/fileexchange/27798-bjontegaard-metric/content/bjontegaard.m
Q33. Z. Wang, H.R. Sheikh, A.C. Bovik, Objective video quality assessment, in *The Handbook of Video Databases: Design and Application*, ed. by B. Furht, O. Marques (CRC press, Sept 2003), pp. 1041–1078
Q34. Y-G. Wang et al., Robust dual watermarking algorithm for AVS video, Signal Process. Image. Commun. **24**,(4), 333–344, (2009)

Real Video 9

R1. Real Networks, Real Video 10—Technical Overview, version 1.0 (2003), (available at http://docs.real.com/docs/rn/rv10/RV10_Tech_Overview.pdf)
R2. KeyLabs Test Final Report—RealNetworks RealVideo 10 Comparison Test (2003) (available at http://docs.real.com/docs/rn/keylabs/RV10Codec.pdf)
R3. Website of Real Networks: http://www.realnetworks.com/

Video Coding Standards

S1. ITU-T Recommendation H.120: Codecs for videoconferencing using primary digital group transmission, Mar 1993

S2. ISO/IEC JTC1/SC29/WG11 MPEG, International standard ISO 11172, Coding of moving pictures and associated audio for digital storage media at up to about 1.5 Mbit/s (1992)

S3. MPEG-2: ISO/IEC 13818-2: Information technology—Generic coding of moving pictures and associated audio information: Video, 1994. ITU-T Recommendation H.262: Information technology—Generic coding of moving pictures and associated audio information: Video, July 1995 ISO/IEC 13818–1: Systems

S4. ITU-T Recommendation H.261: Video codec for audiovisual services at p × 64 kbits/sec, Mar 1993

S5. ITU-T Recommendation H.263: Video coding for low bit rate communication (1998)

S6. Draft ITU-T Recommendation H.263 version 2 (H.263+): video coding for low bitrate communication, Sept 1997; version 3, Nov 2000

S7. G. Côté et al., H.263+: video coding at low bit rates. IEEE Trans. CSVT **8**, 849–866 (1998)

S8. MPEG-4 Standard: ISO/IEC 14496: Information technology—Coding of audio-visual objects (2000)

S9. MPEG-4 Part 2, ISO/IEC 14496-2, Information technology—Coding of audio-visual objects (2001), http://www.iso.ch

S10. MPEG-4: ISO/IEC JTC1/SC29 14496-10: *Information Technology—Coding of Audio-Visual Objects—Part 10: Advanced Video Coding*, 2005. ITU-T Recommendation H.264: Advanced Video Coding for Generic Audiovisual Services, Mar 2003

S11. G.J. Sullivan, *Overview of international video coding standards (preceding H.264/AVC)* (ITU, July, 2005)

S12. Recommendation ITU-R BT.601-7, Studio encoding parameters of digital television for standard 4:3 and wide screen 16:9 aspect ratios, BT Series, Mar 2011

S13. P.N. Tudor, MPEG-2 video compression, IET Electron. Commun. Eng. **7**, 257–264 (1995)

Transcoding

T1. J. Wang et al., An AVS-to-MPEG2 transcoding system, in *IEEE Proceedings of 2004 International Symposium on Intelligent Multimedia, Video and Speech Process., ISIMP'04*, Hong Kong, China, pp. 302–305, Oct 20–22 2004

T2. J. Xin, C. Lin, M.T. Sun, Digital video transcoding. Proc. IEEE **93**, 84–96 (2005)

T3. I. Ahmad et al., Video transcoding: An overview of various techniques and research issues. IEEE Trans. Multimedia **7**, 793–804 (2005)

T4. Z. Wang et al., A fast intra mode decision algorithm for AVS to H.264 transcoding, in *IEEE International Conference on Multimedia and Expo*, Toronto, Canada, pp. 61–64, 9–12 July 2006

T5. M. Wang, G. Sun, Q. Wu, Efficient intra mode decision for AVS to H.264/AVC transcoding, in *IEEE International Symposium on Signal Processing and Information Technology*, Cairo, Egypt, pp. 948–951, Dec 2007

T6. C. Holder, H. Kalva, H.263 to VP6 transcoder. SPIE VCIP, vol. 6822, San Jose, CA, pp. 68222B, Jan 2008

T7. C.A. Segall, J. Zhao, Bit stream rewriting for SVC-to-AVC conversion. IEEE ICIP, San Diego, CA, pp. 2776–2779, Oct 2008

T8. S. Moiron et al., Video transcoding from H.264/AVC to MPEG-2 with reduced complexity. Signal Process. Image Commun. **24**, 637–650 (2009). See also many references. This paper and the references therein can be the basis for various transcoding schemes among different standards MPEG-2, H.264/AVC, AVS China, SMPTE VC-1, DIRAC (BBC) etc.

Video Image Quality (Standards)

U1. ITU-R Rec. BT.500-11, *Methodology for the Subjective Assessment of the Quality of Television Pictures*. (International Telecommunication Union, Geneva, 2002)
U2. ITU-T Rec. P.910, *Subjective Video Quality Assessment Methods for Multimedia Applications*. (International Telecommunication Union, Geneva, 1999)
U3. ITU-T Rec. P.911, *Subjective Audiovisual Quality Assessment Methods for Multimedia Applications*. (International Telecommunication Union, Geneva, 1998)
U4. VQEG, Final report from the video quality experts group on the validation of objective models of video quality assessment (2000), http://www.vqeg.org/
U5. G. Bjontegaard, Calculation of average PSNR differences between RD curves. VCEG-M33, Apr 2001
U6. ITU-T J.144, Objective perceptual video quality measurement techniques for digital cable television in the presence of a full reference (2004)
U7. ITU-T J.149, Method for specifying accuracy and cross-calibration of video quality metrics (VQM) (2004)
U8. ITU-R BT.1683, Objective perceptual video quality measurement techniques for standard definition digital broadcast television in the presence of a full reference (2004)
U9. Video Quality Experts Group (VQEG); VQEG Final Report of FR-TV Phase II Validation Test, Final report from the Video Quality Experts Group on the validation of objective models of video quality assessment, phase II (2003), Available: http://www.its.bldrdoc.gov/vqeg/projects/frtv_phaseII
U10. K. Brunnström et al., VQEG validation and ITU standardization of objective perceptual video quality metrics. IEEE Signal Process. Mag. **26**, 96–101 (2009)
U11. G. Bjontegaard, "Calculation of Average PSNR Differences between RD Curves", ITU-T SC16/Q6, 13th VCEG Meeting, Austin, TX, USA, April 2001, Doc. VCEG-M33
U12. G. Sullivan, Common Conditions for Coding EFFiciency Test, ITU-T Q.15/16, Document VCEG-N81 October 2001. This is a forthcoming revision of ITU-T Q.15/16, Document VCEG-M75 by Gisle Bjontegaard, "Recommended Simulation Conditions for H.26L", 24–27 Sep 2001, Santa barbara, CA, USA
U13. T.K. Tan, G. Sullivan, T. Wedi, Recommended Simulation Conditions for Coding Efficiency Experiments, ITU-T SC16/Q6, 27th VCEG Meeting, Nice, France, 17th–18th October 2005, Doc. VCEG-AA10rl
U14. T.K. Tan, G. Sullivan, T. Wedi, Recommended Simulation Conditions for Coding Efficiency Experiments, ITU-T SC16/Q6, 31st VCEG Meeting, Marrakech, Morocco, 13th–19th January, 2007, Doc. VCEG-AE10rl
U15. T.K. Tan, G. Sullivan, T. Wedi, Recommended Simulation Conditions for Coding Efficiency Experiments, ITU-T SC16/Q6, 34th VCEG Meeting, Marrakech, Antalya, Turkey, 12th–13th January, 2008, Doc. VCEG-AH10r3

Video Coding

V1. P. List et al., Adaptive deblocking filter. IEEE Trans. CSVT **13**, 614–619 (2003)
V2. H. Jia, L. Zhang, Directional diamond search pattern for fast block motion estimation. IEE Electron. Lett. **39**(22), 15811583 (2003)
V3. Video test sequences (YUV 4:2:0): http://trace.eas.asu.edu/yuv/index.html
V4. Video test sequences ITU601: http://www.cipr.rpi.edu/resource/sequences/itu601.html
V5. Link for video sequences: ftp.tnt.uni-hannover.de
V6. K.R. Rao, Mutimedia Processing, Course Website, UT Arlington: http://www-ee.uta.edu/Dip/Courses/EE5359/index.html

V7. DV is a format for recording and playing back of digital video, http://en.wikipedia.org/wiki/DV
V8. MPEG and H.26x series, http://en.wikipedia.org/wiki/MPEG
V9. Interframe coding pictures, Axis Communications. http://www.axis.com/products/video/about_networkvideo/compression.htm
V10. FFmpeg software and official web site: www.ffmpeg.org
V11. Software and source code website for VLC media player, www.videolan.org
V12. Mplayer software and reference web site: http://www.mplayerhq.hu/design7/news.html
V13. Reference software for Aura video converter, www.aura4you.com
V14. Reference software for Matroska media container, www.matroska.org
V15. Test images found in, http://www.hlevkin.com/default.html#testimages

X264 Software

X1. GIT respository of X264—http://git.Videolan.org/?p=x264.git;a=summary
X2. L. Merritt et al., X264: A High Performance H.264/AVC Encoder.
X3. Intel IPP—http://software.intel.com/en-us/articles/intel-intergrated-performance-primitives-code-samples/
X4. FFmpeg software—http://www.ffmpeg.org/
X5. Intel IPP Overview—http://software.intel.com/en-us/intel-ipp/



本书特色

随着多媒体时代的到来以及移动互联网的发展，人们在对于视频的质量要求越来越高的同时，也期望视频传输具有更快的速度。而传输系统和存储系统则要求视频格式乃至码流语法尽可能统一。这两方面的客观要求就是，要对视频数据进行压缩以及压缩要实现标准化，视频编码标准应运而生。K. R. Rao、D. N. Kim和J. J. Hwang合著的本书囊括H.120、H.261、MPEG-1、MPEG-2/H.262、H.263系列、MPEG-4、VP6、Dirac、VC-1、H.264/MPEG-4第10部分、H.265/HEVC和我国的AVS China等从人类进入信息时代至今的主要视频编码标准，并探讨了标准之间的转码问题。本书重点阐述了标准内在的基本功能、工具、技术和操作，涉及的内容之广是同类书罕有的。

本书最具特色，也是最有价值的部分是提供了大量的研究专题，尤其是那些具有前瞻性的尝试和质疑。相信广大读者在充分吸取本书的滋养、挖掘所列专题这一"富矿"之后，对于研究内容的确定是大有裨益的。此外，本书还提供了篇幅可观的附录，其中包含了与专题研究相关的实验数据结果，可作为广大读者自行研究的参考。

地址：北京市百万庄大街22号
邮政编码：100037
电话服务
服务咨询热线：010-88361066
读者购书热线：010-68326294
010-88379203
网络服务
机工官网：www.cmpbook.com
机工官博：weibo.com/cmp1952
金书网：www.golden-book.com
教育服务网：www.cmpedu.com
封面无防伪标均为盗版

机械工业出版社微信公众号
E视界
传播电类内容 提升专业知识
科技电眼
关注电类行业动向 聚焦前沿科技

上架指导 计算机／图像处理
ISBN 978-7-111-56826-1
策划编辑◎吕潇 / 封面设计◎马精明

ISBN 978-7-111-56826-1
9 787111 568261

定价：149.00元